Overview

- Introduction
- Modelling parallel systems
- Linear Time Properties
- Regular Properties
- Linear Temporal Logic (LTL)
- Computation-Tree Logic
- **Equivalences and Abstraction**
  - bisimulation
  - CTL, CTL*-equivalence
  - computing the bisimulation quotient
  - abstraction stutter steps
  - simulation relations ⟵

Classification of implementation relations GRM5.5-CL

Classification of implementation relations

- linear vs. branching time
  - linear time: trace relations
  - branching time: (bi)simulation relations
- (nonsymmetric) preorders vs. equivalences:
  - preorders: trace inclusion, simulation
  - equivalences: trace equivalence, bisimulation
- strong vs. weak relations
  - strong: reasoning about all transitions
  - weak: abstraction from stutter steps

Classification of implementation relations

- linear vs. branching time
  - linear time: trace relations
  - branching time: (bi)simulation relations
- (nonsymmetric) preorders vs. equivalences:
  - preorders: trace inclusion, simulation
  - equivalences: trace equivalence, bisimulation
- strong vs. weak relations
  - strong: reasoning about all transitions
  - weak: abstraction from stutter steps

Classification of implementation relations

- linear vs. branching time
  - linear time: trace relations
  - branching time: (bi)simulation relations
- (nonsymmetric) preorders vs. equivalences:
  - preorders: trace inclusion, simulation
  - equivalences: trace equivalence, bisimulation
- strong vs. weak relations
  - strong: reasoning about all transitions
  - weak: abstraction from stutter steps

The simulation preorder

is a nonsymmetric branching time relation

- plays of central role for abstraction
- the BT-analogue to trace inclusion
- "unidirected" version of bisimulation:

The simulation preorder

is a nonsymmetric branching time relation

- plays of central role for abstraction
- the BT-analogue to trace inclusion
- "unidirected" version of bisimulation:

if $\mathcal{T}_1$ is simulated by $\mathcal{T}_2$ then $\mathcal{T}_2$ can mimick all steps of $\mathcal{T}_1$, but possibly has more behaviors

The simulation preorder

is a nonsymmetric branching time relation

- plays of central role for abstraction
- the BT-analogue to trace inclusion
- "unidirected" version of bisimulation:

if $\mathcal{T}_1$ is simulated by $\mathcal{T}_2$ then $\mathcal{T}_2$ can mimick all steps of $\mathcal{T}_1$, but possibly has more behaviors

- relies on a coinductive definition (as bisimulation equivalence)

here: just strong simulation, i.e., no abstraction from stutter steps

Simulation for two TS

Simulation for two TS

let $\mathcal{T}_1 = (S_1, Act_1, \rightarrow_1, S_{0,1}, AP, L_1)$

$\mathcal{T}_2 = (S_2, Act_2, \rightarrow_2, S_{0,2}, AP, L_2)$

be two transition systems

- over the same set ***AP*** of atomic propositions
- possibly with terminal states

Simulation for a pair of TS

simulation for $(\mathcal{T}_1, \mathcal{T}_2)$: binary relation $\mathcal{R} \ \subseteq \ S_1 \times S_2$ s.t.

(1) if $(s_1, s_2) \in \mathcal{R}$ then $L_1(s_1) = L_2(s_2)$

(2) for all $(s_1, s_2) \in \mathcal{R}$:

$$\forall s_1' \in Post(s_1) \ \exists s_2' \in Post(s_2) \text{ s.t. } (s_1', s_2') \in \mathcal{R}$$

$$\boxed{\begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \\ s_1' & & \end{array}} \quad \text{can be completed to} \quad \boxed{\begin{array}{ccc} s_1 & -\mathcal{R}- & s_2 \\ \downarrow & & \downarrow \\ s_1' & -\mathcal{R}- & s_2' \end{array}}$$

(I) for all initial states s_1 of $\mathcal{T}_1$
there is an initial state s_2 of $\mathcal{T}_2$ with $(s_1, s_2) \in \mathcal{R}$

Simulation preorder $\preceq$

simulation for $(\mathcal{T}_1, \mathcal{T}_2)$: relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

(1) labeling condition

(2) stepwise simulation condition

(I) initial condition

> simulation preorder $\preceq$ for TS:
>
> $$\mathcal{T}_1 \preceq \mathcal{T}_2 \quad \text{iff} \quad \left\{ \begin{array}{c} \text{there exists a simulation } \mathcal{R} \\ \text{for } (\mathcal{T}_1, \mathcal{T}_2) \end{array} \right.$$

Simulation preorder $\preceq$

simulation for $(\mathcal{T}_1, \mathcal{T}_2)$: relation $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

(1) labeling condition

(2) stepwise simulation condition

(I) initial condition

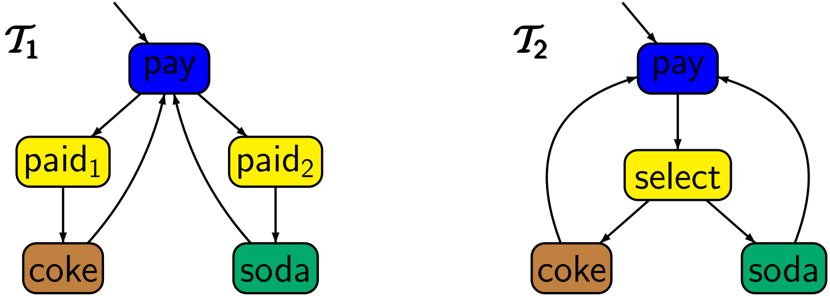
simulation preorder $\preceq$ for TS:

$$\mathcal{T}_1 \preceq \mathcal{T}_2 \quad \text{iff} \quad \left\{ \begin{array}{c} \text{there exists a simulation } \mathcal{R} \\ \text{for } (\mathcal{T}_1, \mathcal{T}_2) \end{array} \right.$$

If s_1 is a state of $\mathcal{T}_1$ and s_2 a state of $\mathcal{T}_2$ then

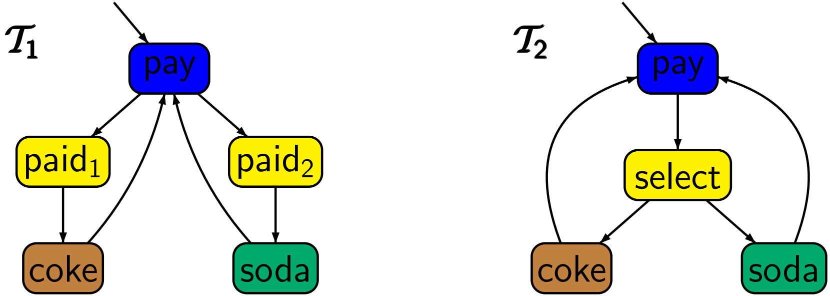
$s_1 \preceq s_2$ iff there exists a simulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$ such that $(s_1, s_2) \in \mathcal{R}$

Two beverage machines



for $AP = \{pay, coke, soda\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$

Two beverage machines

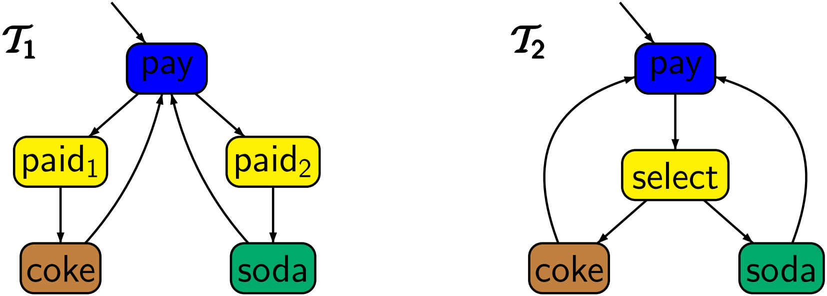


for $AP = \{pay, coke, soda\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$

simulation for $(\mathcal{T}_1, \mathcal{T}_2)$:

$$\{ \ (pay, pay), \ (paid_1, select), \ (paid_2, select), \ (coke, coke), \ (soda, soda) \ \}$$

Two beverage machines

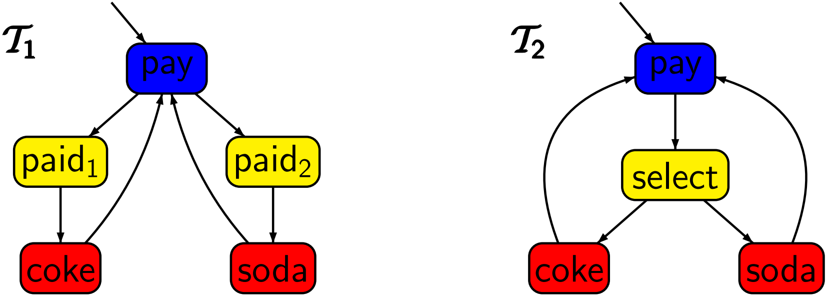


for $AP = \{pay, coke, soda\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$, but $\mathcal{T}_2 \not\preceq \mathcal{T}_1$

simulation for $(\mathcal{T}_1, \mathcal{T}_2)$:

$$\{ \ (pay, pay),\ (paid_1, select),\ (paid_2, select),\ (coke, coke),\ (soda, soda) \ \}$$

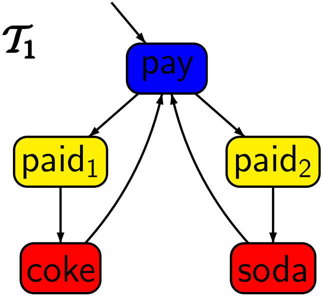
Two beverage machines

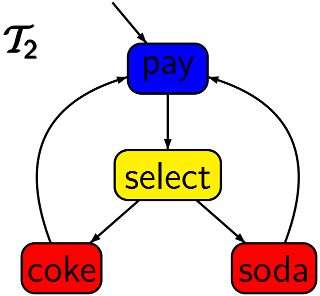


for $AP = \{pay, coke, soda\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$, but $\mathcal{T}_2 \not\preceq \mathcal{T}_1$

for $AP = \{pay, drink\}$:

Two beverage machines

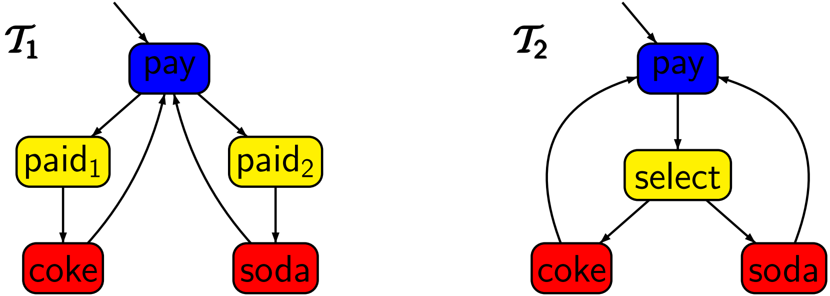




for $AP = \{\textit{pay}, \textit{coke}, \textit{soda}\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$, but $\mathcal{T}_2 \not\preceq \mathcal{T}_1$

for $AP = \{\textit{pay}, \textit{drink}\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$, and $\mathcal{T}_2 \preceq \mathcal{T}_1$

Two beverage machines

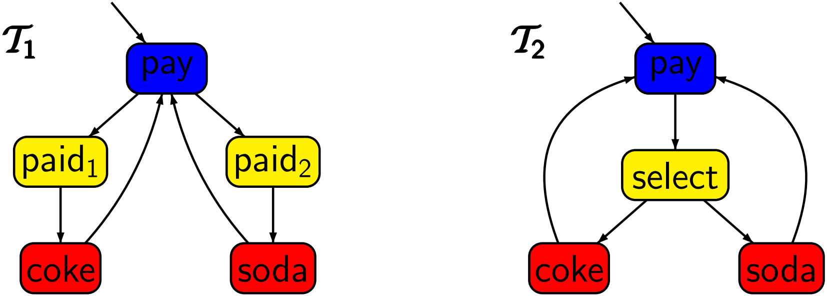


for $AP = \{pay, coke, soda\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$, but $\mathcal{T}_2 \not\preceq \mathcal{T}_1$

for $AP = \{pay, drink\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$, and $\mathcal{T}_2 \preceq \mathcal{T}_1$

simulation for $(\mathcal{T}_1, \mathcal{T}_2)$: as before

Two beverage machines



for $AP = \{pay, coke, soda\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$, but $\mathcal{T}_2 \not\preceq \mathcal{T}_1$

for $AP = \{pay, drink\}$: $\mathcal{T}_1 \preceq \mathcal{T}_2$, and $\mathcal{T}_2 \preceq \mathcal{T}_1$

simulation for $(\mathcal{T}_2, \mathcal{T}_1)$:

$$\{(pay, pay), (select, paid_1), (select, paid_2), (coke, coke), (soda, soda)\}$$

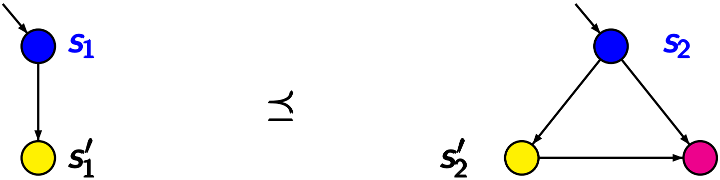
Path fragment lifting for simulation $\mathcal{R}$

$$
\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \\
s_{1,1} & & \\
\downarrow & & \\
s_{1,2} & & \\
\downarrow & & \\
s_{1,3} & & \\
\downarrow & & \\
\vdots & & \\
\downarrow & & \\
s_{1,n} & &
\end{array}
$$

can be completed to

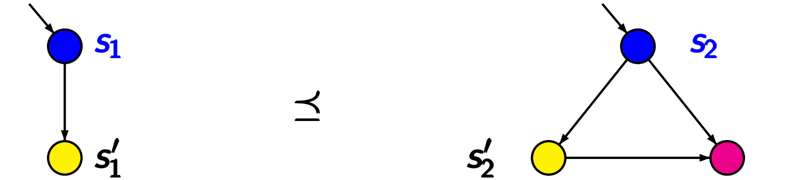
$$
\begin{array}{ccc}
s_1 & -\mathcal{R}- & s_2 \\
\downarrow & & \downarrow \\
s_{1,1} & -\mathcal{R}- & s_{2,1} \\
\downarrow & & \downarrow \\
s_{1,2} & -\mathcal{R}- & s_{2,2} \\
\downarrow & & \downarrow \\
s_{1,3} & -\mathcal{R}- & s_{2,3} \\
\downarrow & & \downarrow \\
\vdots & & \vdots \\
\downarrow & & \downarrow \\
s_{1,n} & -\mathcal{R}- & s_{2,n}
\end{array}
$$

Correct or wrong?

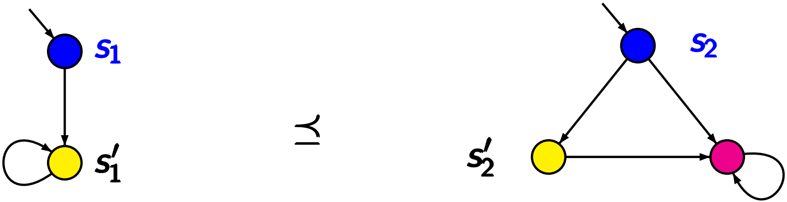


correct. simulation: $\{(s_1, s_2), (s_1', s_2')\}$

Correct or wrong?

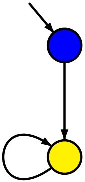


correct. simulation: $\{(s_1, s_2), (s_1', s_2')\}$

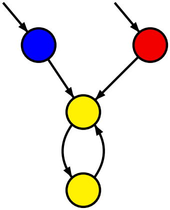


wrong. there is <u>no</u> path fragment in $\mathcal{T}_2$ corresponding to the path fragment $s_1\, s_1'\, s_1'$

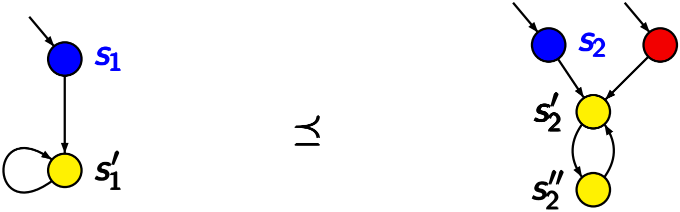
Correct or wrong?



$\preceq$

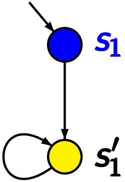


Correct or wrong?

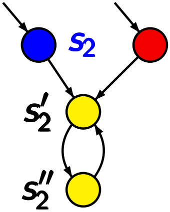


correct. simulation: $\{(s_1, s_2), (s_1', s_2'), (s_1', s_2'')\}$

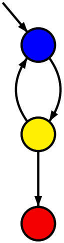
Correct or wrong?



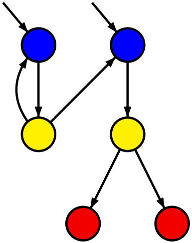
$\preceq$



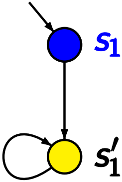
correct. simulation: $\{(s_1, s_2), (s_1', s_2'), (s_1', s_2'')\}$



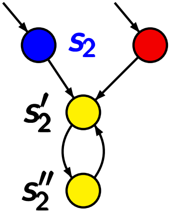
$\preceq$



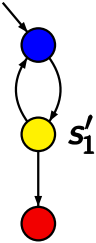
Correct or wrong?



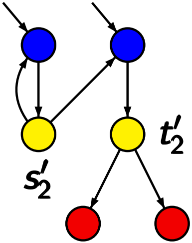
$\preceq$



correct. simulation: $\{(s_1, s_2), (s_1', s_2'), (s_1', s_2'')\}$



$\preceq$



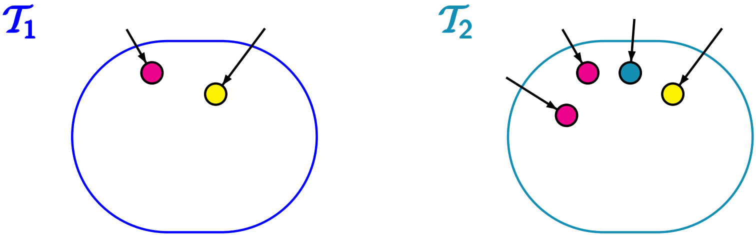
wrong. $s_1' \not\preceq s_2'$ and $s_1' \not\preceq t_2'$

Simulation preorder ...

- as a relation that compares two transition systems

Simulation preorder ...

- as a relation that compares two transition systems

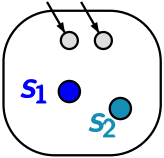


Simulation preorder ...

- as a relation that compares two transition systems
- as a relation on the states of one transition system

- as a relation that compares two transition systems
- as a relation on the states of one transition system

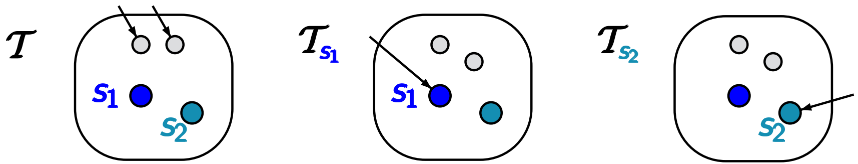
$\mathcal{T}$



$s_1 \preceq_{\mathcal{T}} s_2$ iff ?

Simulation preorder ...

- as a relation that compares two transition systems
- as a relation on the states of one transition system



$s_1 \preceq_{\mathcal{T}} s_2$	iff	$\mathcal{T}_{s_1} \preceq \mathcal{T}_{s_2}$
	iff	there exists a simulation $\mathcal{R}$ for $\mathcal{T}$ with $(s_1, s_2) \in \mathcal{R}$

Simulation preorder for a single TS

Let $\mathcal{T} = (S, Act, \rightarrow, \ldots)$ be a transition system.

The simulation preorder $\preceq_{\mathcal{T}}$ is the coarsest relation on S such that for all states $s_1, s_2 \in S$ with $s_1 \preceq_{\mathcal{T}} s_2$:

Simulation preorder for a single TS

Let $\mathcal{T} = (S, Act, \rightarrow, \ldots)$ be a transition system.

The simulation preorder $\preceq_{\mathcal{T}}$ is the coarsest relation on S such that for all states $s_1, s_2 \in S$ with $s_1 \preceq_{\mathcal{T}} s_2$:

(1) $L(s_1) = L(s_2)$

(2) each transition of s_1 can be mimicked by a transition of s_2

$$\begin{tikzcd} s_1 \arrow[d] & \preceq_{\mathcal{T}} & s_2 \\ s_1' & & \end{tikzcd}$$

can be completed to

$$\begin{tikzcd} s_1 \arrow[d] & \preceq_{\mathcal{T}} & s_2 \arrow[d] \\ s_1' & \preceq_{\mathcal{T}} & s_2' \end{tikzcd}$$

Simulation preorder for a single TS

Let $\mathcal{T} = (S, Act, \rightarrow, \ldots)$ be a transition system.

The simulation preorder $\preceq_{\mathcal{T}}$ is the coarsest relation on S such that for all states $s_1, s_2 \in S$ with $s_1 \preceq_{\mathcal{T}} s_2$:

(1) $L(s_1) = L(s_2)$

(2) each transition of s_1 can be mimicked by a transition of s_2

$\preceq_{\mathcal{T}}$ is a preorder, i.e., transitive and reflexive.

Let $\mathcal{T}$ be a transition system with state space S.

A simulation for $\mathcal{T}$ is a binary relation $\mathcal{R} \subseteq S \times S$ s.t.

(1) if $(s_1, s_2) \in \mathcal{R}$ then $L(s_1) = L(s_2)$

(2) for all $(s_1, s_2) \in \mathcal{R}$:

$$\forall s_1' \in Post(s_1)\, \exists s_2' \in Post(s_2) \text{ s.t. } (s_1', s_2') \in \mathcal{R}$$

simulation preorder $\preceq_{\mathcal{T}}$:

$s_1 \preceq_{\mathcal{T}} s_2$ iff there exists a simulation $\mathcal{R}$ for $\mathcal{T}$ s.t. $(s_1, s_2) \in \mathcal{R}$

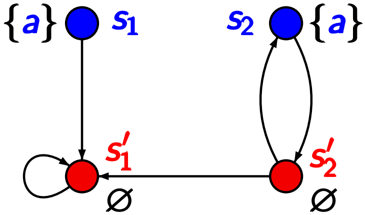
Path fragment lifting for $\preceq_{\mathcal{T}}$

$$
\begin{array}{ccc}
s_1 & \preceq_{\mathcal{T}} & s_2 \\
\downarrow & & \\
s_{1,1} & & \\
\downarrow & & \\
s_{1,2} & & \\
\downarrow & & \\
s_{1,3} & & \\
\downarrow & & \\
\vdots & & \\
\downarrow & & \\
s_{1,n} & &
\end{array}
$$

can be completed to

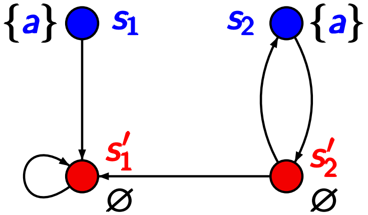
$$
\begin{array}{ccc}
s_1 & \preceq_{\mathcal{T}} & s_2 \\
\downarrow & & \downarrow \\
s_{1,1} & \preceq_{\mathcal{T}} & s_{2,1} \\
\downarrow & & \downarrow \\
s_{1,2} & \preceq_{\mathcal{T}} & s_{2,2} \\
\downarrow & & \downarrow \\
s_{1,3} & \preceq_{\mathcal{T}} & s_{2,3} \\
\downarrow & & \downarrow \\
\vdots & & \vdots \\
\downarrow & & \downarrow \\
s_{1,n} & \preceq_{\mathcal{T}} & s_{2,n}
\end{array}
$$

Example: simulation preorder $\preceq_{\mathcal{T}}$



$s_1 \ \preceq_{\mathcal{T}} \ s_2$

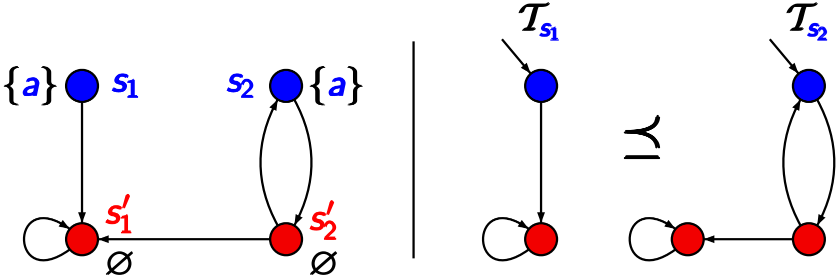
Example: simulation preorder $\preceq_{\mathcal{T}}$



$s_1 \preceq_{\mathcal{T}} s_2$ as

$\big\{(s_1, s_2), (s_1', s_2'), (s_1', s_1')\big\}$ is a simulation for $\mathcal{T}$

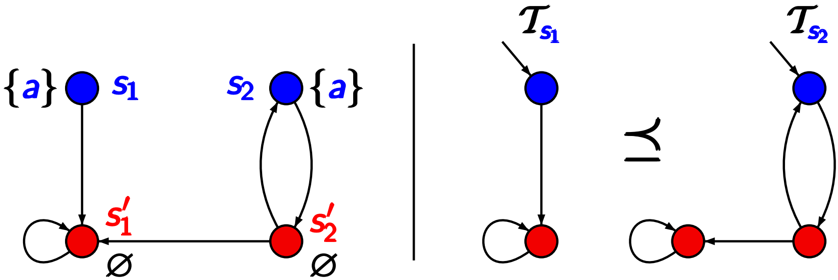
Example: simulation preorder $\preceq_{\mathcal{T}}$



$s_1 \preceq_{\mathcal{T}} s_2$ as

$\{(s_1, s_2), (s_1', s_2'), (s_1', s_1')\}$ is a simulation for $\mathcal{T}$

Example: simulation preorder $\preceq_{\mathcal{T}}$



$s_1 \preceq_{\mathcal{T}} s_2$ as

$\{(s_1, s_2), (s_1', s_2'), (s_1', s_1')\}$ is a simulation for $\mathcal{T}$

$$s_1 \to s_1' \to s_1' \to s_1' \to \ldots$$

is simulated by

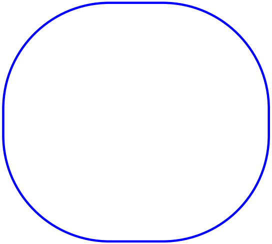
$$s_2 \to s_2' \to s_1' \to s_1' \to \ldots$$

Abstraction and simulation

GRM5.5-6

Abstraction and simulation

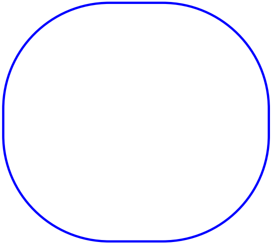
GRM5.5-6



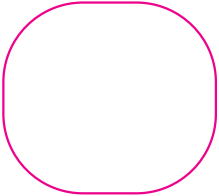
transition system $\mathcal{T}$
with state space S

Abstraction and simulation

GRM5.5-6



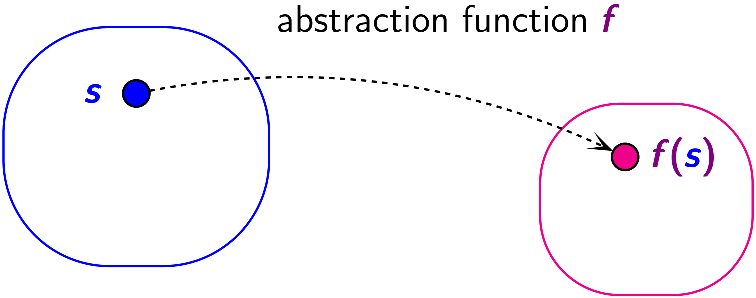
transition system $\mathcal{T}$
with state space S



"small" abstract
state space S'

Abstraction and simulation

GRM5.5-6

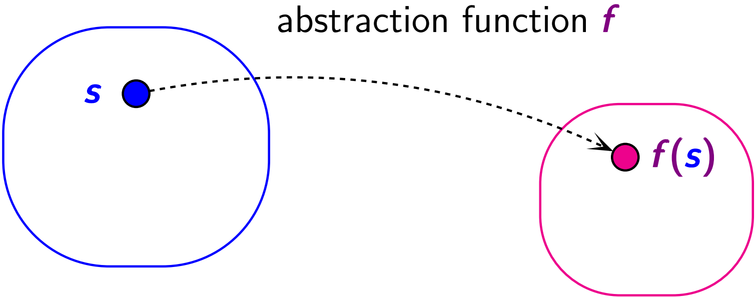


transition system $\mathcal{T}$
with state space S

abstract transition system
$\mathcal{T}_f$ with state space S'

Abstraction and simulation

GRM5.5-6



transition system $\mathcal{T}$ with state space S

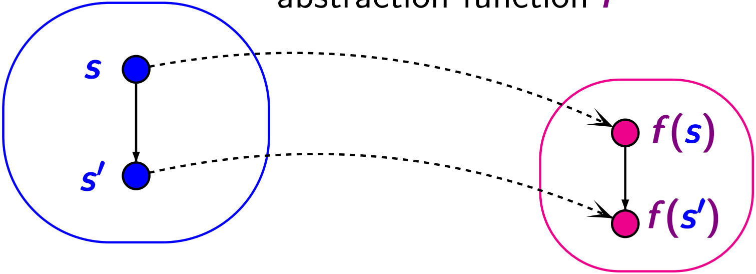
abstract transition system $\mathcal{T}_f$ with state space S'

lifting of transitions:

$$\frac{s \longrightarrow s'}{f(s) \longrightarrow f(s')}$$

Abstraction and simulation

abstraction function f



transition system $\mathcal{T}$ with state space S

abstract transition system $\mathcal{T}_f$ with state space S'

lifting of transitions:

$$\frac{s \longrightarrow s'}{f(s) \longrightarrow f(s')}$$

Abstraction and simulation

given: transition system $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$

set S' and abstraction function $f : S \to S'$

s.t. $L(s) = L(t)$ if $f(s) = f(t)$ for all $s, t \in S$

given: transition system $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$

set S' and abstraction function $f : S \to S'$

s.t. $L(s) = L(t)$ if $f(s) = f(t)$ for all $s, t \in S$

goal: define abstract transition system $\mathcal{T}_f$

with state space S' s.t. $\mathcal{T} \preceq \mathcal{T}_f$

Abstraction and simulation

abstraction function $f : S \to S'$ s.t.

$L(s) = L(t)$ if $f(s) = f(t)$ for all $s, t \in S$

transition system

$$\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$$

$$\Downarrow$$

abstract transition system

$$\mathcal{T}_f = (S', Act', \longrightarrow_f, S'_0, AP, L')$$

abstraction function $f : S \to S'$ s.t.
$L(s) = L(t)$ if $f(s) = f(t)$ for all $s, t \in S$

transition system
$$\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$$
$$\Downarrow$$
abstract transition system
$$\mathcal{T}_f = (S', Act', \longrightarrow_f, S'_0, AP, L')$$

where $S'_0 = \{ f(s_0) : s_0 \in S_0 \}$ and $L'(f(s)) = L(s)$

$$\frac{s \longrightarrow s'}{f(s) \longrightarrow_f f(s')}$$

abstraction function $f : S \to S'$ s.t.

$L(s) = L(t)$ if $f(s) = f(t)$ for all $s, t \in S$

$$\boxed{\begin{array}{c}\text{transition system}\\ \mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)\\ \Downarrow \\ \text{abstract transition system}\\ \mathcal{T}_f = (S', Act', \longrightarrow_f, S'_0, AP, L')\end{array}}$$

Then $\mathcal{T} \preceq \mathcal{T}_f$

Abstraction and simulation

abstraction function $f : S \to S'$ s.t.

$L(s) = L(t)$ if $f(s) = f(t)$ for all $s, t \in S$

transition system

$$\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$$

$$\Downarrow$$

abstract transition system

$$\mathcal{T}_f = (S', Act', \longrightarrow_f, S_0', AP, L')$$

Then $\mathcal{T} \preceq \mathcal{T}_f$ $\longleftarrow$ $\mathcal{R} = \{\langle s, f(s)\rangle : s \in S\}$ is a simulation for $(\mathcal{T}, \mathcal{T}_f)$

```
WHILE  x > 0 DO
   x := x−1;
   y := y+1
OD
IF  even(y)
  THEN  return "1"
  ELSE  return "0"
FI
```

$$x \in \mathbb{N}$$
$$y \in \mathbb{N}$$

Data abstraction

```
WHILE x > 0 DO
   x := x−1;
   y := y+1
OD
IF even(y)
  THEN return "1"
  ELSE return "0"
FI
```

$\xrightarrow{\text{data abstr.}}$

$$x \in \mathbb{N} \quad \longrightarrow \quad x \in \{gzero, zero\}$$

$$y \in \mathbb{N} \quad \longrightarrow \quad y \in \{even, odd\}$$

```
WHILE x > 0 DO
   x := x-1;
   y := y+1
OD
IF even(y)
  THEN return "1"
  ELSE return "0"
FI
```

data abstr. $\longrightarrow$

```
WHILE x = gzero DO
   x := gzero or x := zero

  IF y = even
     THEN y := odd
     ELSE y := even
  FI
OD
IF y = even
   THEN return "1"
   ELSE return "0"
FI
```

$x \in \mathbb{N} \quad \longrightarrow \quad x \in \{gzero, zero\}$

$y \in \mathbb{N} \quad \longrightarrow \quad y \in \{even, odd\}$

Data abstraction

WHILE $x > \mathbf{0}$ DO
  $x := x - 1$;
  $\boxed{y := y + 1}$
OD
IF $even(y)$
  THEN return "**1**"
  ELSE return "**0**"
FI

data abstr. $\longrightarrow$

WHILE $x = gzero$ DO
  $x := gzero$ or $x := zero$
  IF $y = even$
    THEN $y := odd$
    ELSE $y := even$
  FI
OD
IF $y = even$
  THEN return "**1**"
  ELSE return "**0**"
FI

concrete operation $\rightsquigarrow$ abstract operation

Data abstraction

WHILE $x > 0$ DO
  $\boxed{x := x-1;}$
  $y := y+1$
OD
IF $even(y)$
  THEN return "1"
  ELSE return "0"
FI

data abstr. $\longrightarrow$

WHILE $x = gzero$ DO
  $\boxed{x := gzero \texttt{ or } x := zero}$
  IF $y = even$
    THEN $y := odd$
    ELSE $y := even$
  FI
OD
IF $y = even$
  THEN return "1"
  ELSE return "0"
FI

concrete operation $\rightsquigarrow$ abstract operation, e.g.,

$x := x-1$ $\qquad$ $gzero \mapsto gzero \texttt{ or } zero$

Abstraction and simulation

GRM5.5-8

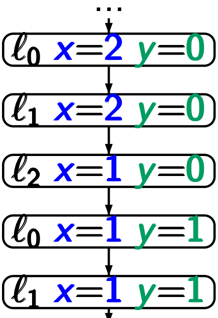
abstract TS simulates the concrete one

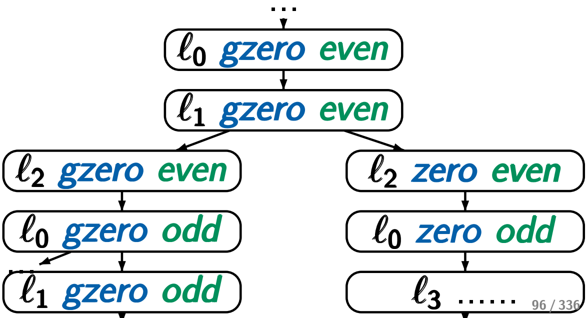
```
WHILE x > 0 DO
   x := x-1
   y := y+1
OD

IF even(y)
 THEN return 1
 ELSE return 0
```

```
WHILE x = gzero DO
   x := gzero or x := zero
   IF y = even
        THEN y := odd
        ELSE y := even
   FI
OD
IF y = even
     THEN return 1
     ELSE return 0  FI
```

```
ℓ0  WHILE x > 0 DO
ℓ1     x := x−1
ℓ2     y := y+1
    OD

ℓ3  IF even(y)
ℓ4   THEN return 1
ℓ5   ELSE return 0
```

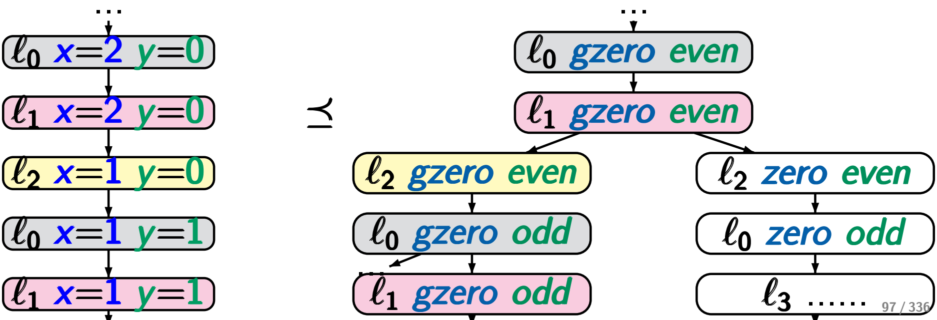
```
ℓ0  WHILE x = gzero DO
ℓ1     x := gzero or x := zero
ℓ2     IF y = even
              THEN y := odd
              ELSE y := even
       FI
    OD
ℓ3  IF y = even
ℓ4        THEN return 1
ℓ5        ELSE return 0   FI
```





```
ℓ0  WHILE  x > 0 DO
ℓ1      x := x−1
ℓ2      y := y+1
    OD

ℓ3  IF  even(y)
ℓ4   THEN  return 1
ℓ5   ELSE  return 0
```

```
ℓ0  WHILE  x = gzero DO
ℓ1     x := gzero or x := zero
ℓ2     IF  y = even
              THEN  y := odd
              ELSE  y := even
       FI
    OD
ℓ3  IF  y = even
ℓ4        THEN  return 1
ℓ5        ELSE  return 0   FI
```



Simulation preorder vs. and trace inclusion

$$\boxed{\mathcal{T}_1 \preceq \mathcal{T}_2 \quad \Longrightarrow \quad \mathit{Tracesfin}(\mathcal{T}_1) \subseteq \mathit{Tracesfin}(\mathcal{T}_2)}$$

reason: path fragment lifting for $\preceq$

Simulation preorder vs. and trace inclusion

$$\mathcal{T}_1 \preceq \mathcal{T}_2 \implies \mathit{Tracesfin}(\mathcal{T}_1) \subseteq \mathit{Tracesfin}(\mathcal{T}_2)$$

if $\mathcal{T}_1$ does not have terminal states, then:

$$\mathcal{T}_1 \preceq \mathcal{T}_2 \implies \mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$$

... does not hold if $\mathcal{T}_1$ has terminal states ...

$\mathcal{T}_1 \preceq \mathcal{T}_2$

$$\mathit{Traces}(\mathcal{T}_1) = \{\varnothing\,\varnothing\} \neq \{\varnothing^\omega\} = \mathit{Traces}(\mathcal{T}_2)$$

kernel of the simulation preorder, i.e.,

$$\simeq \; = \; \preceq \cap \preceq^{-1}$$

For TS $\mathcal{T}_1$ and $\mathcal{T}_2$ over the same set of atomic propositions:

$$\mathcal{T}_1 \simeq \mathcal{T}_2 \quad \text{iff} \quad \mathcal{T}_1 \preceq \mathcal{T}_2 \text{ and } \mathcal{T}_2 \preceq \mathcal{T}_1$$

Simulation equivalence $\simeq_{\mathcal{T}}$

kernel of the simulation preorder, i.e.,

$$\simeq \; = \; \preceq \cap \preceq^{-1}$$

For TS $\mathcal{T}_1$ and $\mathcal{T}_2$ over the same set of atomic propositions:

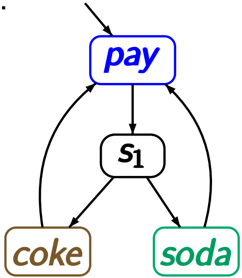
$$\mathcal{T}_1 \simeq \mathcal{T}_2 \quad \text{iff} \quad \mathcal{T}_1 \preceq \mathcal{T}_2 \text{ and } \mathcal{T}_2 \preceq \mathcal{T}_1$$

for states s_1 and s_2 of a TS $\mathcal{T}$:

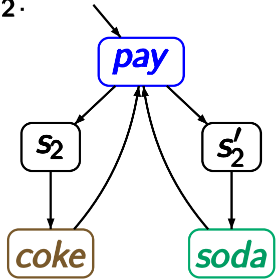
$$s_1 \simeq_{\mathcal{T}} s_2 \quad \text{iff} \quad s_1 \preceq_{\mathcal{T}} s_2 \text{ and } s_2 \preceq_{\mathcal{T}} s_1$$

Two beverage machines

$\mathcal{T}_1$:

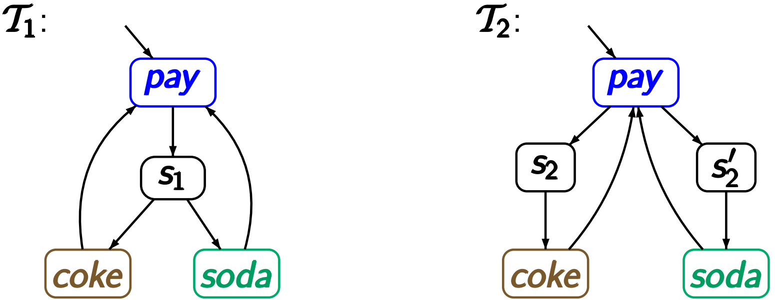


$\mathcal{T}_2$:



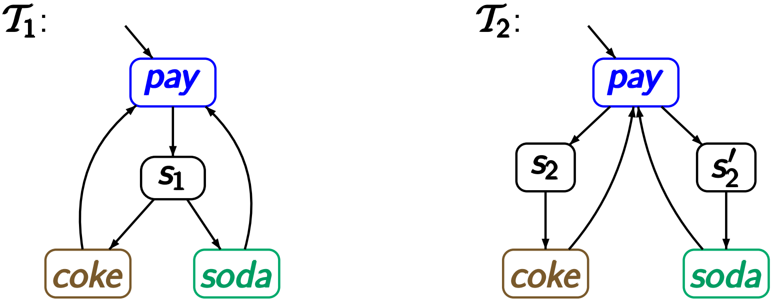
for $AP = \{pay, coke, soda\}$

Two beverage machines



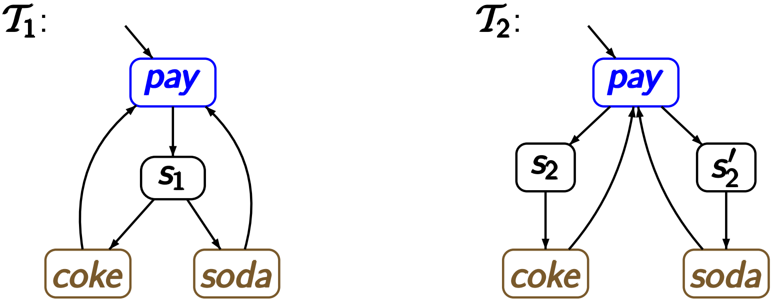
for $AP = \{pay, coke, soda\}$

$\mathcal{T}_2 \preceq \mathcal{T}_1$, but $\mathcal{T}_1 \not\preceq \mathcal{T}_2$



for $AP = \{pay, coke, soda\}$

$\mathcal{T}_2 \preceq \mathcal{T}_1$, but $\mathcal{T}_1 \not\simeq \mathcal{T}_2$ $\longleftarrow$ $\boxed{\text{since } \mathcal{T}_1 \not\preceq \mathcal{T}_2}$



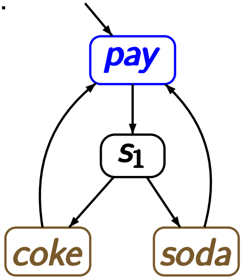
for $AP = \{pay, coke, soda\}$

$\mathcal{T}_2 \preceq \mathcal{T}_1$, but $\mathcal{T}_1 \not\simeq \mathcal{T}_2$ $\longleftarrow$ $\boxed{\text{since } \mathcal{T}_1 \not\preceq \mathcal{T}_2}$

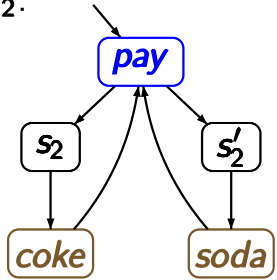
for $AP = \{pay, drink\}$:

Two beverage machines

$\mathcal{T}_1$:



$\mathcal{T}_2$:

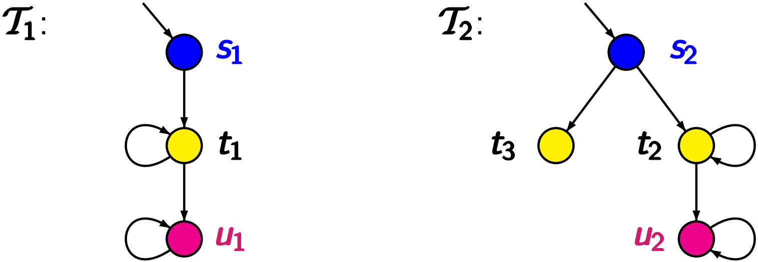


for $AP = \{pay, coke, soda\}$

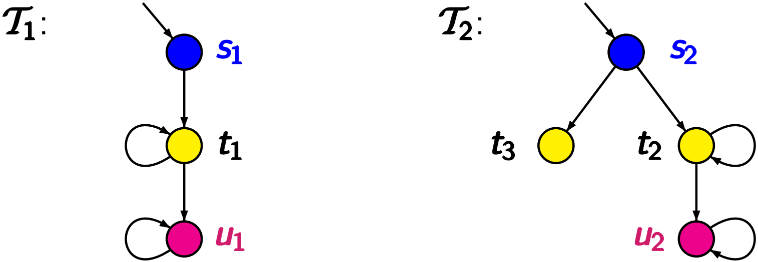
$\mathcal{T}_2 \preceq \mathcal{T}_1$, but $\mathcal{T}_1 \not\simeq \mathcal{T}_2 \longleftarrow$ $\boxed{\text{since } \mathcal{T}_1 \not\preceq \mathcal{T}_2}$

for $AP = \{pay, drink\}$: $\mathcal{T}_1 \simeq \mathcal{T}_2$

Example: simulation equivalent TS



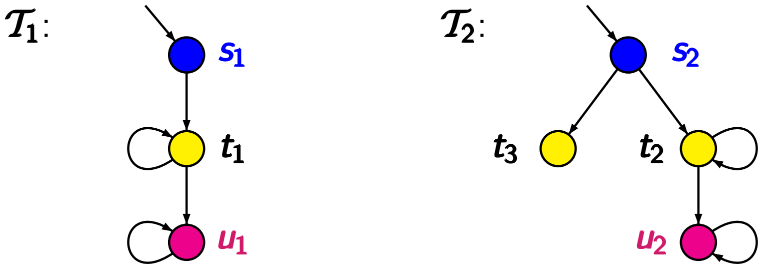
Example: simulation equivalent TS



simulation for $(\mathcal{T}_1, \mathcal{T}_2)$:

$$\{(s_1, s_2), (t_1, t_2), (u_1, u_2)\}$$

Example: simulation equivalent TS



simulation for $(\mathcal{T}_1, \mathcal{T}_2)$:

$$\{(s_1, s_2), (t_1, t_2), (u_1, u_2)\}$$

simulation for $(\mathcal{T}_2, \mathcal{T}_1)$:

$$\{(s_2, s_1), (t_2, t_1), (t_3, t_1), (u_2, u_1)\}$$

Bisimulation vs. simulation equivalence

Bisimulation equivalence $\sim$ is strictly finer than simulation equivalence $\simeq$

Bisimulation vs. simulation equivalence

Bisimulation equivalence $\sim$ is strictly finer than simulation equivalence $\simeq$

That is:

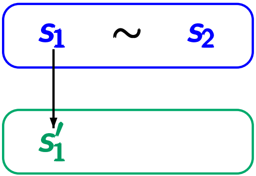
1. $\mathcal{T}_1 \sim \mathcal{T}_2$ implies $\mathcal{T}_1 \simeq \mathcal{T}_2$

   Proof: Let $\mathcal{R}$ is a bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$.

   - $\mathcal{R}$ is a simulation for $(\mathcal{T}_1, \mathcal{T}_2) \implies \mathcal{T}_1 \preceq \mathcal{T}_2$
   - $\mathcal{R}^{-1}$ is a simulation for $(\mathcal{T}_2, \mathcal{T}_1) \implies \mathcal{T}_2 \preceq \mathcal{T}_1$

2. there exist TS $\mathcal{T}_1$ and $\mathcal{T}_2$ s.t. $\mathcal{T}_1 \simeq \mathcal{T}_2$ and $\mathcal{T}_1 \not\sim \mathcal{T}_2$

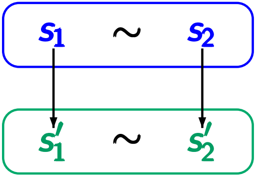
bisimulation equivalence



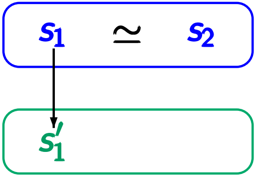
bisimulation equivalence

$$\begin{tikzcd} s_1 \arrow[d] \arrow[r, phantom, "\sim"] & s_2 \arrow[d] \\ s_1' \arrow[r, phantom, "\sim"] & s_2' \end{tikzcd}$$

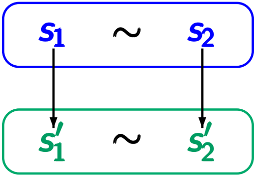
bisimulation equivalence



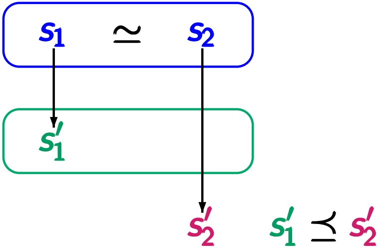
simulation equivalence



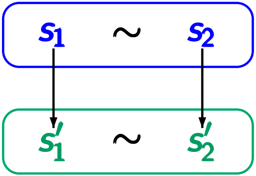
bisimulation equivalence



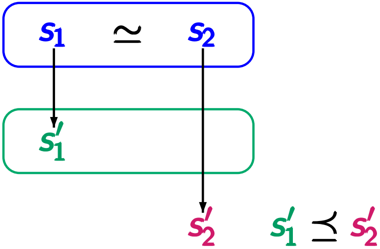
simulation equivalence



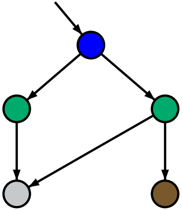
bisimulation equivalence

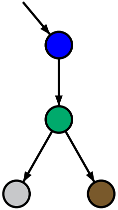


simulation equivalence



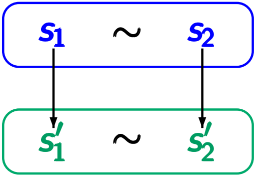
$\mathcal{T}_1$



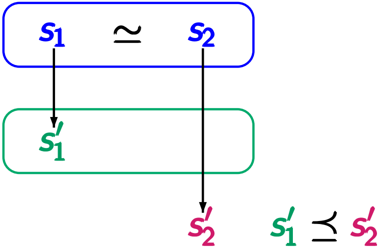


$\mathcal{T}_2$

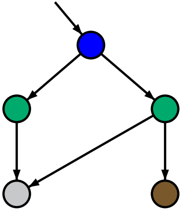
bisimulation equivalence



simulation equivalence

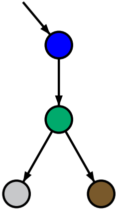


$\mathcal{T}_1$

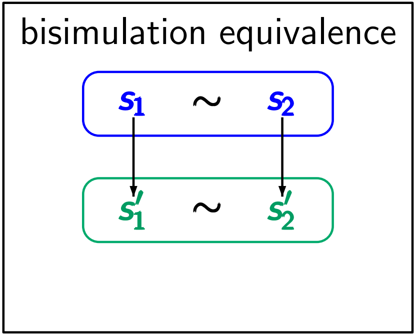


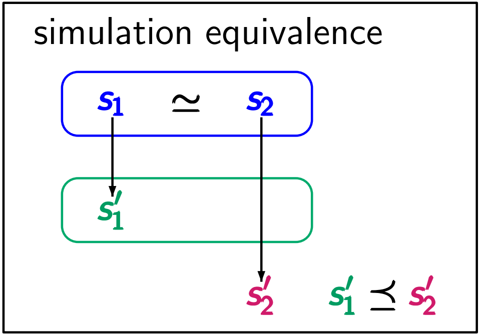
$\simeq$

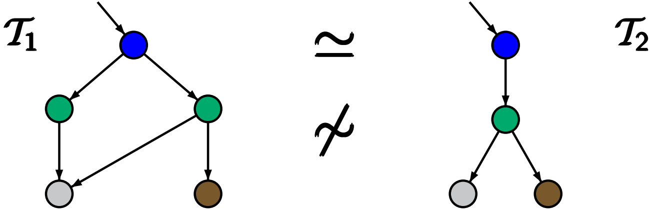
$\not\sim$



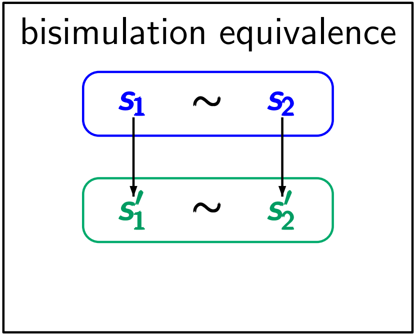
$\mathcal{T}_2$

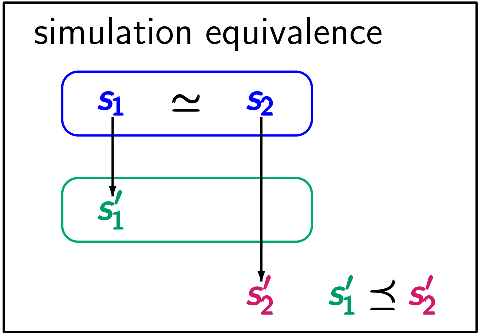


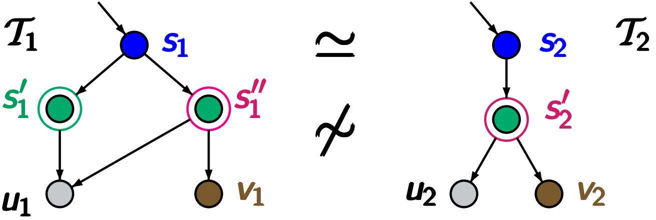




$\mathcal{T}_2 \preceq \mathcal{T}_1$, as $\mathcal{T}_2$ is a "subsystem" of $\mathcal{T}_1$







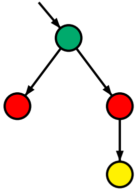
simulation for $(\mathcal{T}_1, \mathcal{T}_2)$:

$$\{(s_1, s_2), (s_1', s_2'), (s_1'', s_2'), (u_1, u_2), (v_1, v_2)\}$$

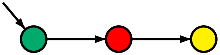
Simulation vs trace equivalence

$$\mathcal{T}_1 \simeq \mathcal{T}_2 \quad \not\Longrightarrow \quad \mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$$

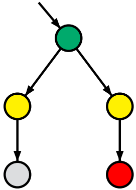
$$\mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2) \quad \not\Longrightarrow \quad \mathcal{T}_1 \simeq \mathcal{T}_2$$



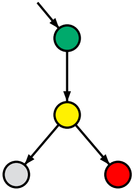
$\simeq$



not trace equivalent
but simulation equivalent



$\not\simeq$

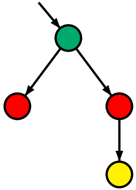


trace equivalent
not simulation equivalent

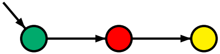
Simulation vs trace equivalence $\longleftarrow$ incomparable

$$\mathcal{T}_1 \simeq \mathcal{T}_2 \quad \not\Rightarrow \quad Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$$

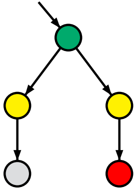
$$Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2) \quad \not\Rightarrow \quad \mathcal{T}_1 \simeq \mathcal{T}_2$$



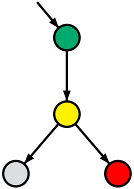
$\simeq$



not trace equivalent
but simulation equivalent



$\not\simeq$



trace equivalent
not simulation equivalent

Simulation vs. finite trace equivalence

$$\mathcal{T}_1 \simeq \mathcal{T}_2 \quad \not\Longrightarrow \quad Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$$

$$Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2) \quad \not\Longrightarrow \quad \mathcal{T}_1 \simeq \mathcal{T}_2$$

$$\mathcal{T}_1 \simeq \mathcal{T}_2 \quad \Longrightarrow \quad Tracesfin(\mathcal{T}_1) = Tracesfin(\mathcal{T}_2)$$

while "$\Longleftarrow$" does not hold

Simulation vs. finite trace equivalence

$$\mathcal{T}_1 \simeq \mathcal{T}_2 \quad \not\Rightarrow \quad \mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$$

$$\mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2) \quad \not\Rightarrow \quad \mathcal{T}_1 \simeq \mathcal{T}_2$$

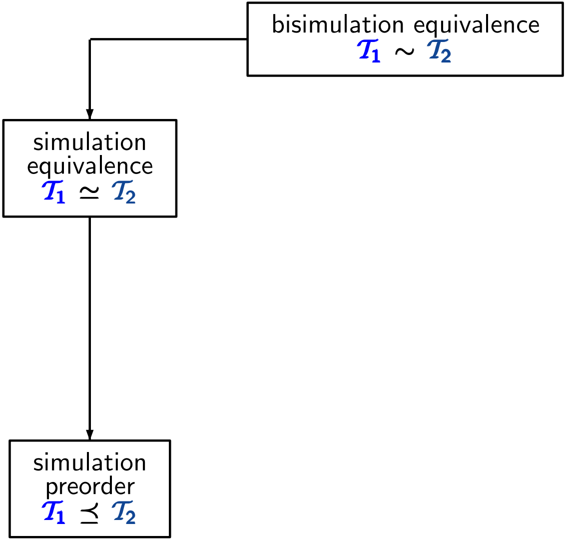
$$\mathcal{T}_1 \simeq \mathcal{T}_2 \quad \Longrightarrow \quad \mathit{Tracesfin}(\mathcal{T}_1) = \mathit{Tracesfin}(\mathcal{T}_2)$$

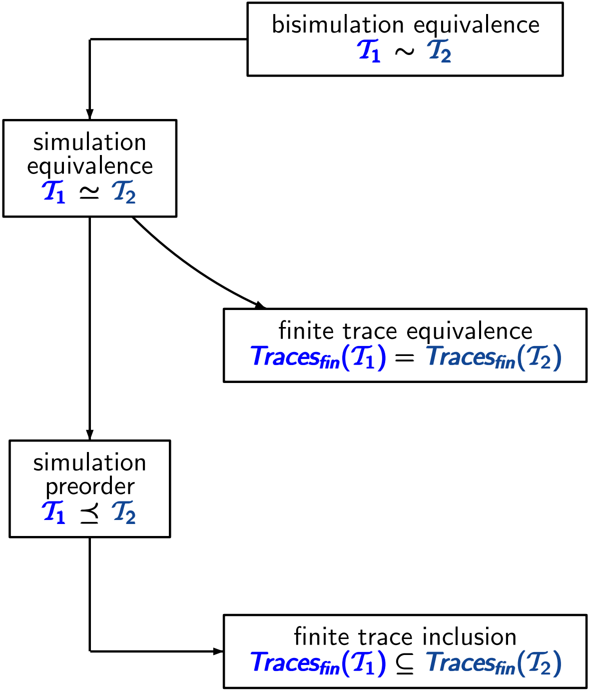
while "$\Longleftarrow$" does not hold

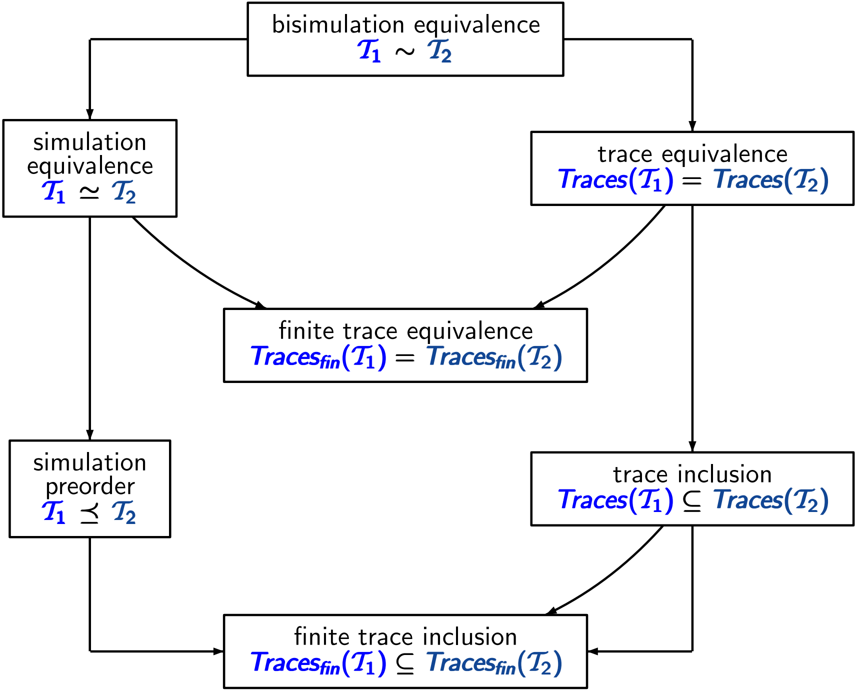
If $\mathcal{T}_1$, $\mathcal{T}_2$ do not have terminal states then:

$$\mathcal{T}_1 \simeq \mathcal{T}_2 \quad \Longrightarrow \quad \mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$$

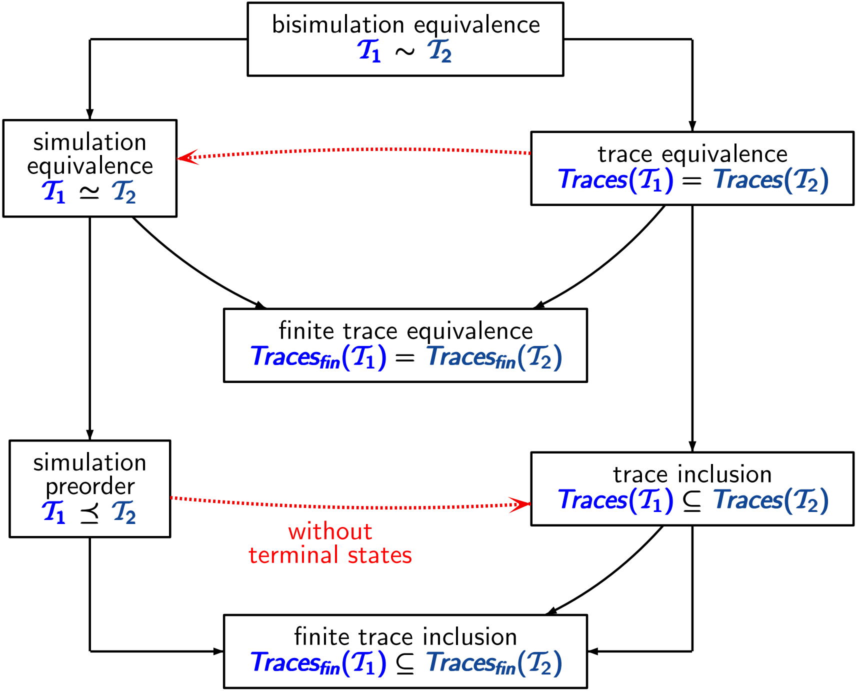
Summary: trace and (bi)simulation relations

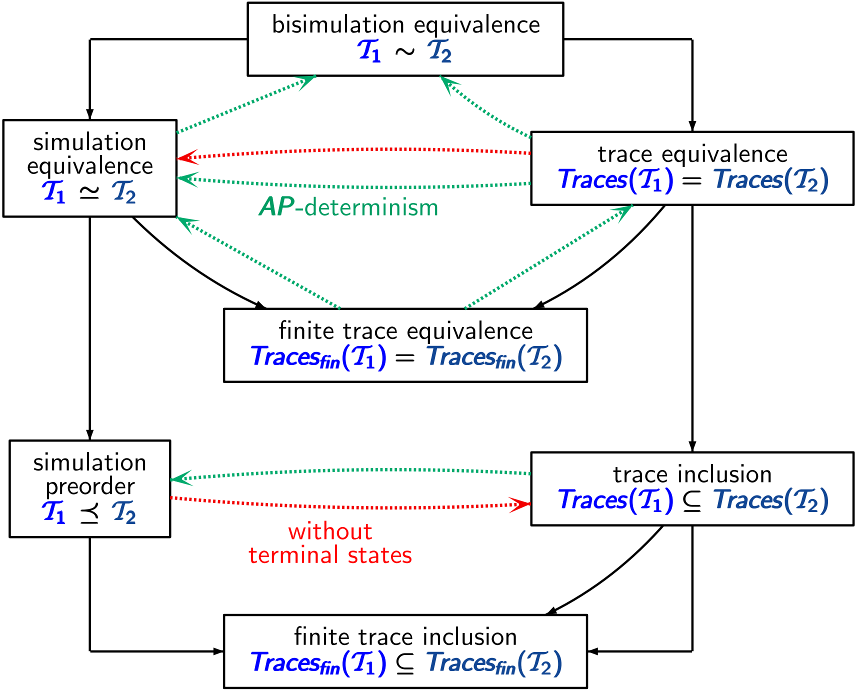

bisimulation equivalence
$\mathcal{T}_1 \sim \mathcal{T}_2$
simulation equivalence
$\mathcal{T}_1 \simeq \mathcal{T}_2$
simulation preorder
$\mathcal{T}_1 \preceq \mathcal{T}_2$


bisimulation equivalence
$\mathcal{T}_1 \sim \mathcal{T}_2$
simulation equivalence
$\mathcal{T}_1 \simeq \mathcal{T}_2$
finite trace equivalence
$Traces_{fin}(\mathcal{T}_1) = Traces_{fin}(\mathcal{T}_2)$
simulation preorder
$\mathcal{T}_1 \preceq \mathcal{T}_2$
finite trace inclusion
$Traces_{fin}(\mathcal{T}_1) \subseteq Traces_{fin}(\mathcal{T}_2)$



bisimulation equivalence
$\mathcal{T}_1 \sim \mathcal{T}_2$
simulation equivalence
$\mathcal{T}_1 \simeq \mathcal{T}_2$
trace equivalence
$Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$
finite trace equivalence
$Traces_{fin}(\mathcal{T}_1) = Traces_{fin}(\mathcal{T}_2)$
simulation preorder
$\mathcal{T}_1 \preceq \mathcal{T}_2$
trace inclusion
$Traces(\mathcal{T}_1) \subseteq Traces(\mathcal{T}_2)$
finite trace inclusion
$Traces_{fin}(\mathcal{T}_1) \subseteq Traces_{fin}(\mathcal{T}_2)$


bisimulation equivalence
$\mathcal{T}_1 \sim \mathcal{T}_2$
simulation equivalence
$\mathcal{T}_1 \simeq \mathcal{T}_2$
trace equivalence
$Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$
finite trace equivalence
$Traces_{fin}(\mathcal{T}_1) = Traces_{fin}(\mathcal{T}_2)$
simulation preorder
$\mathcal{T}_1 \preceq \mathcal{T}_2$
trace inclusion
$Traces(\mathcal{T}_1) \subseteq Traces(\mathcal{T}_2)$
without terminal states
finite trace inclusion
$Traces_{fin}(\mathcal{T}_1) \subseteq Traces_{fin}(\mathcal{T}_2)$



bisimulation equivalence
$\mathcal{T}_1 \sim \mathcal{T}_2$
simulation equivalence
$\mathcal{T}_1 \simeq \mathcal{T}_2$
trace equivalence
$Traces(\mathcal{T}_1) = Traces(\mathcal{T}_2)$
AP-determinism
finite trace equivalence
$Traces_{fin}(\mathcal{T}_1) = Traces_{fin}(\mathcal{T}_2)$
simulation preorder
$\mathcal{T}_1 \preceq \mathcal{T}_2$
trace inclusion
$Traces(\mathcal{T}_1) \subseteq Traces(\mathcal{T}_2)$
without terminal states
finite trace inclusion
$Traces_{fin}(\mathcal{T}_1) \subseteq Traces_{fin}(\mathcal{T}_2)$

AP-determinism

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

$\mathcal{T}$ is called AP-deterministic iff

(1) for all states s and all subsets A of AP:

$$\left|\{ t \in S : s \rightarrow t \wedge L(t) = A \}\right| \leqslant 1$$

(2) for all subsets A of AP:

$$\left|\{ s_0 \in S_0 : L(s_0) = A \}\right| \leqslant 1$$

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$.

$$\boxed{\begin{array}{l}\text{If } Traces_{fin}(s_1) = Traces_{fin}(s_2) \text{ then} \\ \qquad Traces(s_1) = Traces(s_2)\end{array}}$$

mainly because:

- each (finite or infinite) word σ_1 over 2^{AP} is induced by at most one path fragment starting in s_1 or s_2, respectively
- if $\sigma = A_0 A_1 \ldots A_i A_{i+1} \ldots \in Traces(s_1)$ then there is no proper prefix $A_0 A_1 \ldots A_i$ of σ belongs to $Traces(s_1)$

  $+$ analogous statement for s_2

Correct or wrong?

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$.

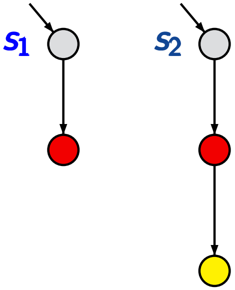
If $\mathit{Traces}_{\mathit{fin}}(s_1) \subseteq \mathit{Traces}_{\mathit{fin}}(s_2)$ then

$$\mathit{Traces}(s_1) \subseteq \mathit{Traces}(s_2)$$

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$.

$$\boxed{\begin{array}{l}\text{If } Traces_{fin}(s_1) \subseteq Traces_{fin}(s_2) \text{ then} \\ \qquad Traces(s_1) \subseteq Traces(s_2)\end{array}}$$

wrong.



$Traces_{fin}(s_1) \subseteq Traces_{fin}(s_2)$

 $\in Traces(s_1) \setminus Traces(s_2)$

(Bi)simulation and trace equivalence

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$. Then the following statements are equivalent:

(1)	$s_1 \sim_{\mathcal{T}} s_2$	(bisimulation equivalence)
(2)	$s_1 \simeq_{\mathcal{T}} s_2$	(simulation equivalence)
(3)	$Traces_{fin}(s_1) = Traces_{fin}(s_2)$	
(4)	$Traces(s_1) = Traces(s_2)$	

(1) $\Longrightarrow$ (2): $\checkmark$

(2) $\Longrightarrow$ (3): ... path fragment lifting ...

(3) $\Longrightarrow$ (4): just shown

(4) $\Longrightarrow$ (1): ...

Bisimulation and trace equivalence

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$. Then:

$$Traces(s_1) = Traces(s_2) \text{ implies } s_1 \sim_{\mathcal{T}} s_2$$

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$. Then:

$$Traces(s_1) = Traces(s_2) \text{ implies } s_1 \sim_{\mathcal{T}} s_2$$

Proof: show that

$$\mathcal{R} = \{(s_1, s_2) : Traces(s_1) = Traces(s_2)\}$$

is a bisimulation.

Bisimulation and trace equivalence

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$. Then:

$$Traces(s_1) = Traces(s_2) \text{ implies } s_1 \sim_{\mathcal{T}} s_2$$

Proof: show that

$$\mathcal{R} = \{(s_1, s_2) : Traces(s_1) = Traces(s_2)\}$$

is a bisimulation.

Note that if $s \rightarrow t$ then

Bisimulation and trace equivalence

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$. Then:

$$Traces(s_1) = Traces(s_2) \text{ implies } s_1 \sim_{\mathcal{T}} s_2$$

Proof: show that

$$\mathcal{R} = \{(s_1, s_2) : Traces(s_1) = Traces(s_2)\}$$

is a bisimulation.

Note that if $s \rightarrow t$ then

$$Traces(t) = \{L(t)B_1B_2B_3\ldots \in (2^{AP})^+ \cup (2^{AP})^\omega : L(s)L(t)B_1B_2B_3\ldots \in Traces(s)\}$$

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$. Then:

$$Traces_{fin}(s_1) = Traces_{fin}(s_2) \text{ implies } s_1 \sim_{\mathcal{T}} s_2$$

Let $\mathcal{T}$ be AP-deterministic and s_1, s_2 states in $\mathcal{T}$. Then:

$$Traces_{fin}(s_1) = Traces_{fin}(s_2) \text{ implies } s_1 \sim_{\mathcal{T}} s_2$$

Proof: show that

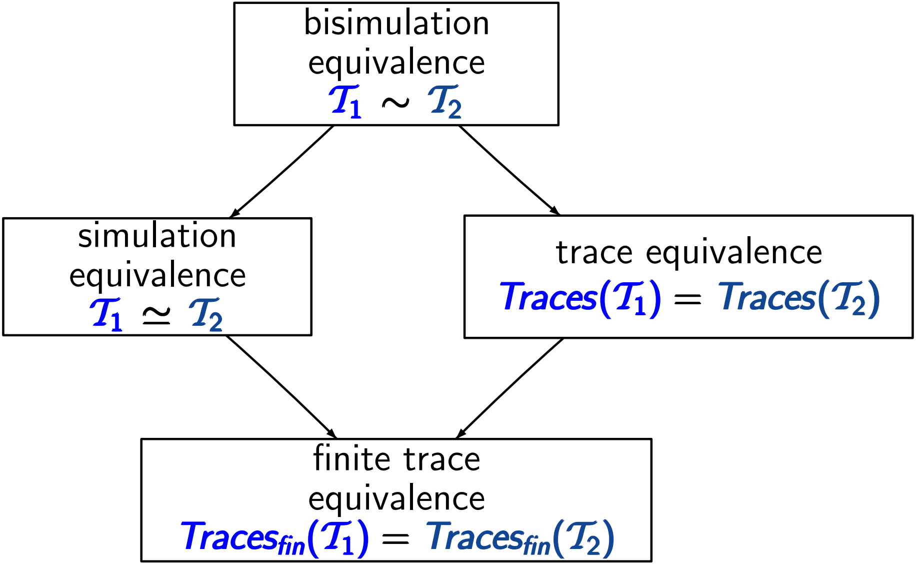
$$\mathcal{R} = \{(s_1, s_2) : Traces_{fin}(s_1) = Traces_{fin}(s_2)\}$$

is a bisimulation.

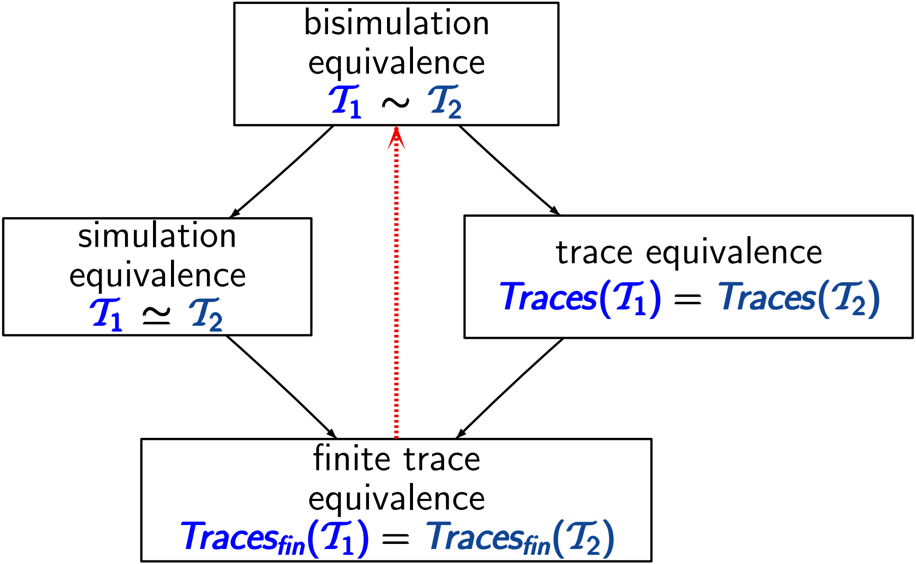
Note that if $s \rightarrow t$ then

$$Traces_{fin}(t) = \{L(t)B_1B_2\ldots B_n \in (2^{AP})^+ : L(s)L(t)B_1B_2\ldots B_n \in Traces_{fin}(s)\}$$

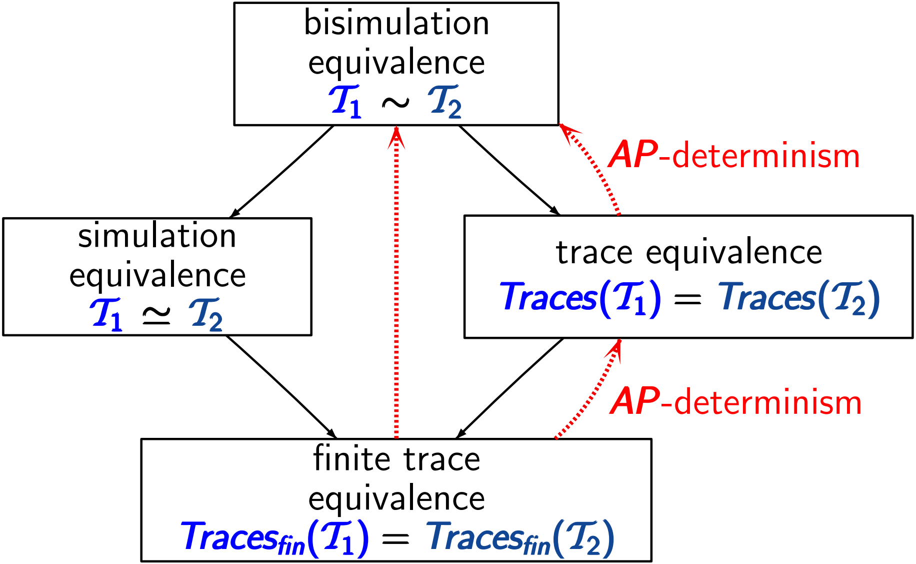
Trace and (bi)simulation equivalence



For AP-deterministic TS

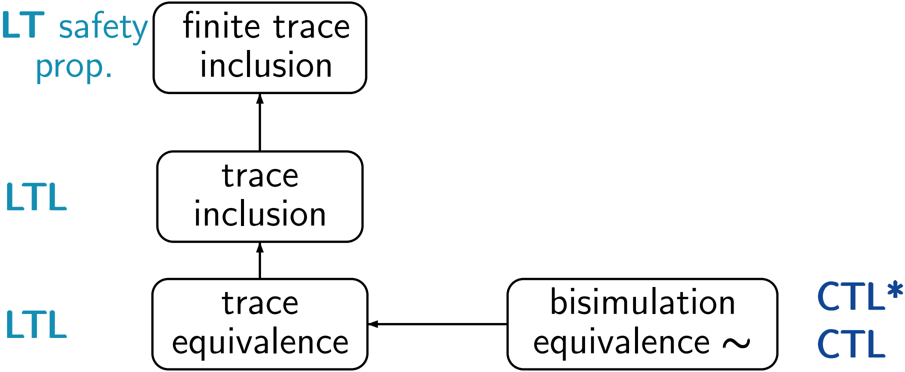


For AP-deterministic TS

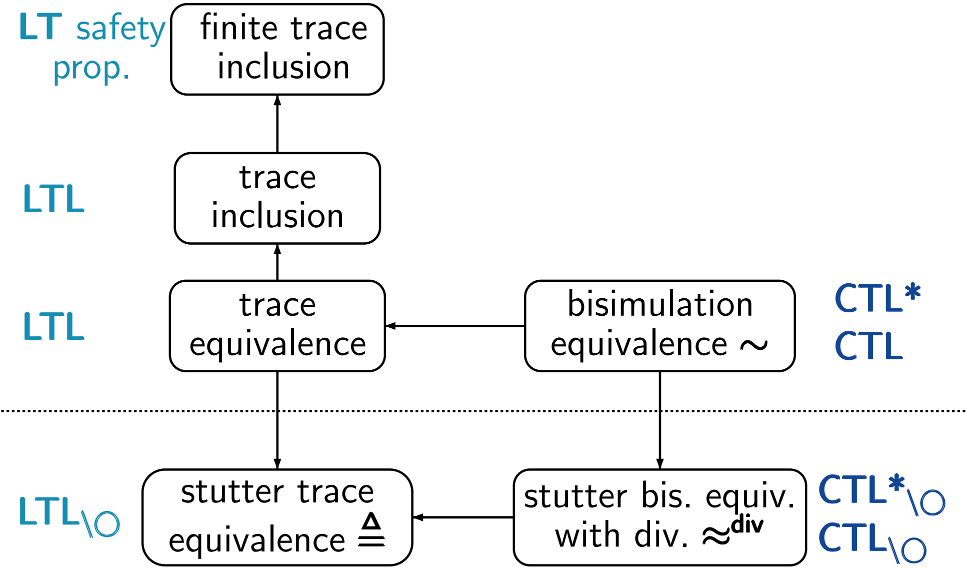


Logical characterizations

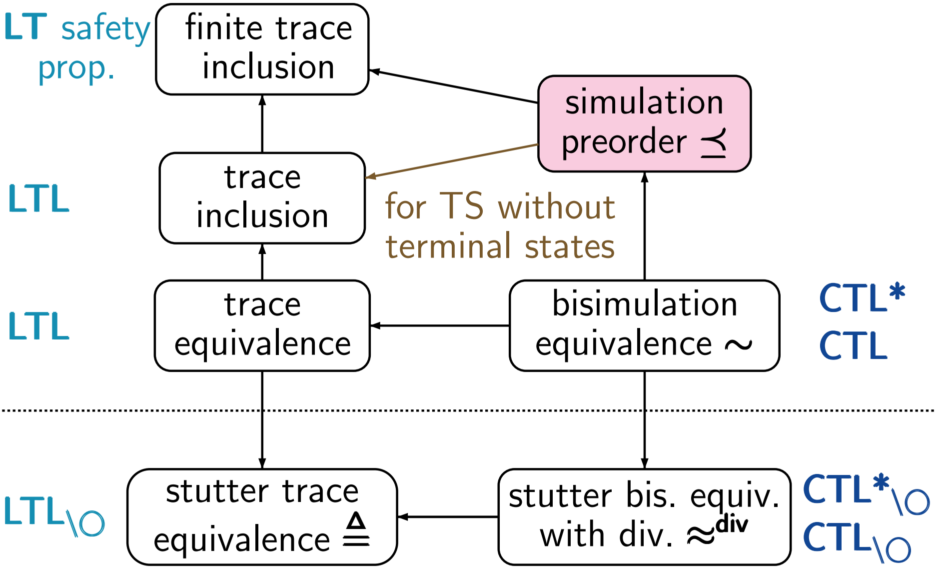
GRM5.5-19



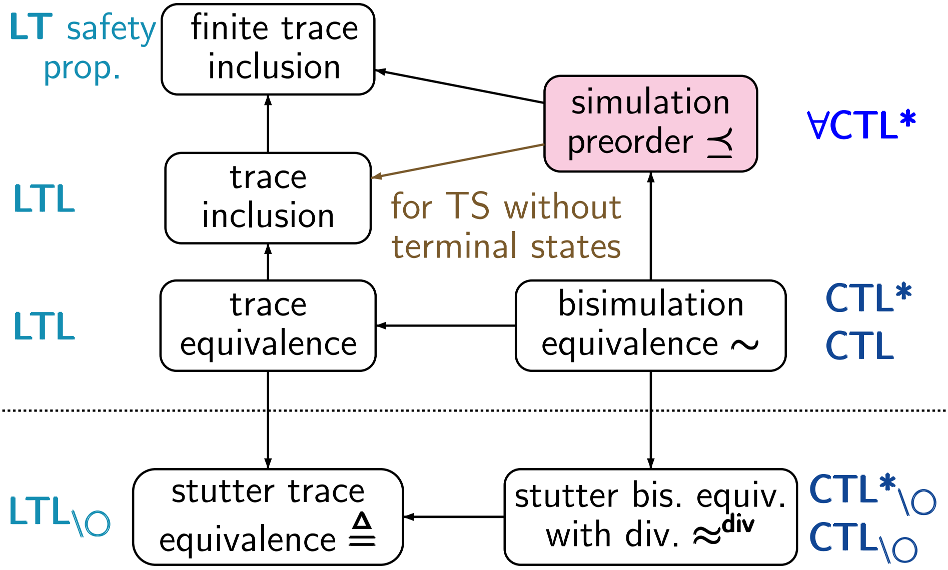
Logical characterizations



Logical characterizations



Logical characterizations



Logical characterization

for bisimulation equivalence $\sim_{\mathcal{T}}$:

$s_1 \sim_{\mathcal{T}} s_2$ iff s_1, s_2 satisfy the same **CTL*** formulas
iff s_1, s_2 satisfy the same **CTL** formulas

Logical characterization

for bisimulation equivalence $\sim_{\mathcal{T}}$:

$s_1 \sim_{\mathcal{T}} s_2$	iff	s_1, s_2 satisfy the same **CTL*** formulas
	iff	s_1, s_2 satisfy the same **CTL** formulas

for the simulation preorder $\preceq_{\mathcal{T}}$:

by a sublogic $\mathbb{L}$ of **CTL*** that subsumes **LTL**

$s_1 \preceq_{\mathcal{T}} s_2$ iff for all formulas $\Phi \in \mathbb{L}$:

$$s_2 \models \Phi \text{ implies } s_1 \models \Phi$$

observation: $\mathbb{L}$ <u>cannot</u> be closed under negation

The universal fragment $\forall$CTL* of CTL*

CTL* formulas in positive normal form, without $\exists$

Syntax of ∀CTL*

∀CTL* state formulas:

$$\Phi \quad ::= \quad \mathit{true} \;\mid\; \mathit{false} \;\mid\; a \;\mid\; \neg a \;\mid\; \Phi_1 \wedge \Phi_2 \;\mid\; \Phi_1 \vee \Phi_2 \;\mid\; \forall \varphi$$

∀CTL* path formulas:

$$\varphi \quad ::= \quad \Phi \;\mid\; \varphi_1 \wedge \varphi_2 \;\mid\; \varphi_1 \vee \varphi_2 \;\mid\; \bigcirc \varphi \;\mid\; \varphi_1 \,\mathsf{U}\, \varphi_2 \;\mid\; \varphi_1 \,\mathsf{W}\, \varphi_2$$

Syntax of ∀CTL*

∀CTL* state formulas:

$$\Phi \;::=\; \mathit{true} \;\mid\; \mathit{false} \;\mid\; a \;\mid\; \neg a \;\mid\; \Phi_1 \wedge \Phi_2 \;\mid\; \Phi_1 \vee \Phi_2 \;\mid\; \forall \varphi$$

∀CTL* path formulas:

$$\varphi \;::=\; \Phi \;\mid\; \varphi_1 \wedge \varphi_2 \;\mid\; \varphi_1 \vee \varphi_2 \;\mid\; \bigcirc \varphi \;\mid\; \varphi_1 \,\mathsf{U}\, \varphi_2 \;\mid\; \varphi_1 \,\mathsf{W}\, \varphi_2$$

eventually: $\Diamond \varphi \stackrel{\text{def}}{=} \mathit{true}\,\mathsf{U}\,\varphi$

always: $\Box \varphi \stackrel{\text{def}}{=} \varphi\,\mathsf{W}\,\mathit{false}$

Embedding of LTL in ∀CTL*

∀CTL* state formulas:

$$\Phi \quad ::= \quad \mathit{true} \;\mid\; \mathit{false} \;\mid\; a \;\mid\; \neg a \;\mid$$
$$\Phi_1 \wedge \Phi_2 \;\mid\; \Phi_1 \vee \Phi_2 \;\mid\; \forall \varphi$$

∀CTL* path formulas:

$$\varphi \quad ::= \quad \Phi \;\mid\; \varphi_1 \wedge \varphi_2 \;\mid\; \varphi_1 \vee \varphi_2 \;\mid\; \bigcirc \varphi \;\mid$$
$$\varphi_1 \,\mathsf{U}\, \varphi_2 \;\mid\; \varphi_1 \,\mathsf{W}\, \varphi_2$$

for all **LTL** formulas φ in **PNF**:

$$s \models_{\mathrm{LTL}} \varphi \quad \text{iff} \quad s \models_{\forall\mathrm{CTL}^*} \forall \varphi$$

but $\forall\Diamond\forall\Box a$ cannot be expressed in **LTL**

The universal fragments of CTL* and CTL

syntax of $\forall$**CTL***:

$$\Phi ::= \mathit{true} \mid \mathit{false} \mid a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \forall \varphi$$

$$\varphi ::= \Phi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc \varphi \mid \varphi_1 \, \mathbf{U} \, \varphi_2 \mid \varphi_1 \, \mathbf{W} \, \varphi_2$$

$\forall$**CTL**: sublogic of $\forall$**CTL***

- no Boolean operators for paths formulas
- the arguments of the temporal modalities $\bigcirc$, **U** and **W** are state formulas

The universal fragments of CTL* and CTL

syntax of $\forall$**CTL***:

$$\Phi \;::=\; \mathit{true} \mid \mathit{false} \mid a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \forall \varphi$$

$$\varphi \;::=\; \Phi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc \varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \mid \varphi_1 \,\mathsf{W}\, \varphi_2$$

$\forall$**CTL**: sublogic of $\forall$**CTL***

syntax of $\forall$**CTL**:

$$\Phi \;::=\; \mathit{true} \mid \mathit{false} \mid a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \forall \bigcirc \Phi \mid \forall(\Phi_1 \,\mathsf{U}\, \Phi_2) \mid \forall(\Phi_1 \,\mathsf{W}\, \Phi_2)$$

Logical characterization of simulation

Let $\mathcal{T}$ be a finite TS without terminal states. Then, for all states s_1 and s_2 in $\mathcal{T}$, the following statements are equivalent:

(1) $s_1 \preceq_{\mathcal{T}} s_2$

(2) for all $\forall$**CTL** state formulas Φ:

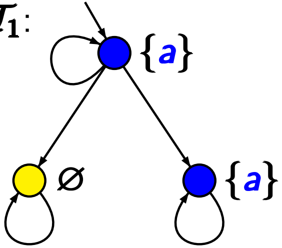
$$\text{if } s_2 \models \Phi \text{ then } s_1 \models \Phi$$

(3) for all $\forall$**CTL*** state formulas Φ:

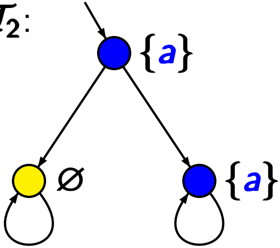
$$\text{if } s_2 \models \Phi \text{ then } s_1 \models \Phi$$

∀CTL and simulation

$\mathcal{T}_1$:

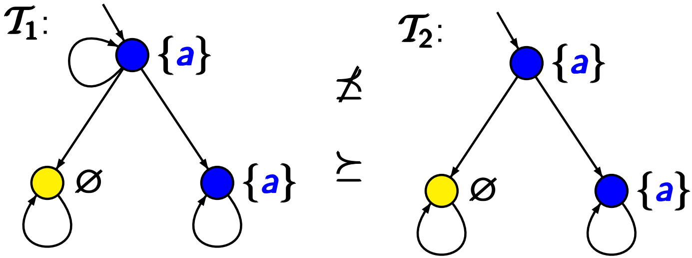


$\mathcal{T}_2$:



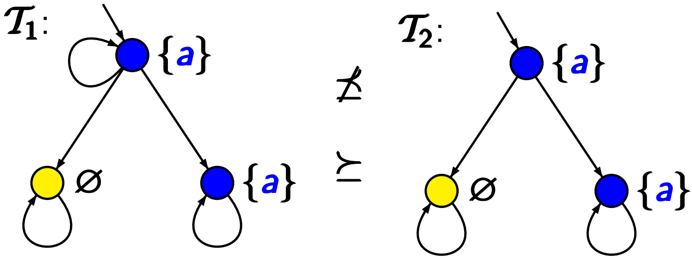
$AP = \{a\}$

$\forall$CTL and simulation



$AP = \{a\}$

∀CTL and simulation



$AP = \{a\}$

e.g., $\mathcal{T}_1 \not\models \forall\bigcirc(\forall\bigcirc\neg a \ \vee \ \forall\bigcirc a)$

$\mathcal{T}_2 \models \forall\bigcirc(\forall\bigcirc\neg a \ \vee \ \forall\bigcirc a)$

$\mathcal{T}_1 \not\models \forall\Diamond(\forall\Box\neg a \ \vee \ \forall\Box a)$

$\mathcal{T}_2 \models \forall\Diamond(\forall\Box\neg a \ \vee \ \forall\Box a)$

$\forall$CTL/$\forall$CTL* and the simulation preorder

For finite TS without terminal states, the following statements are equivalent:

(1) $s_1 \preceq_{\mathcal{T}} s_2$

(2) for all **$\forall$CTL** formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$

(3) for all **$\forall$CTL*** formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$

$\forall$CTL/$\forall$CTL* and the simulation preorder

For finite TS without terminal states, the following statements are equivalent:

(1)	$s_1 \preceq_{\mathcal{T}} s_2$
(2)	for all **$\forall$CTL** formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$
(3)	for all **$\forall$CTL*** formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$

(3) $\Longrightarrow$ (2): obvious as **$\forall$CTL** is a sublogic of **$\forall$CTL***

$\forall$CTL/$\forall$CTL* and the simulation preorder

For finite TS without terminal states, the following statements are equivalent:

(1) $s_1 \preceq_{\mathcal{T}} s_2$

(2) for all **$\forall$CTL** formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$

(3) for all **$\forall$CTL*** formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$

(3) $\Longrightarrow$ (2): obvious as **$\forall$CTL** is a sublogic of **$\forall$CTL***

(1) $\Longrightarrow$ (3): holds for arbitrary (possibly infinite) TS without terminal states

proof by structural induction

$\forall$CTL/$\forall$CTL* and the simulation preorder

For finite TS without terminal states, the following statements are equivalent:

(1)	$s_1 \preceq_{\mathcal{T}} s_2$
(2)	for all $\forall$CTL formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$
(3)	for all $\forall$CTL* formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$

(1) $\Longrightarrow$ (3): show by structural induction:

(i) for all $\forall$CTL* state formulas Φ and states s_1, s_2:
if $s_1 \preceq_{\mathcal{T}} s_2$ and $s_2 \models \Phi$ then $s_1 \models \Phi$

(ii) for all $\forall$CTL* path formulas φ and paths π_1, π_2:
if $\pi_1 \preceq_{\mathcal{T}} \pi_2$ and $\pi_2 \models \varphi$ then $\pi_1 \models \varphi$

$\forall$CTL/$\forall$CTL* and the simulation preorder

For finite TS without terminal states, the following statements are equivalent:

(1)	$s_1 \preceq_{\mathcal{T}} s_2$
(2)	for all $\forall$CTL formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$
(3)	for all $\forall$CTL* formulas Φ: $s_2 \models \Phi$ implies $s_1 \models \Phi$

(2) $\Longrightarrow$ (1): show that for finite TS:

$$\mathcal{R} = \Big\{ (s_1, s_2) \;:\; \text{for all } \forall\text{CTL formulas } \Phi: \; s_2 \models \Phi \text{ implies } s_1 \models \Phi \Big\}$$

is a simulation.

Duality of ∀CTL* and ∃CTL*

∃CTL* (state) formulas:

$$\Psi \quad ::= \quad \mathit{true} \mid \mathit{false} \mid a \mid \neg a \mid \Psi_1 \wedge \Psi_2 \mid \Psi_1 \vee \Psi_2 \mid \exists\varphi$$

∃CTL* path formulas:

$$\varphi \quad ::= \quad \Psi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \mid \varphi_1 \,\mathsf{W}\, \varphi_2$$

analogous: ∃CTL

For each ∀CTL* formula Φ there is a ∃CTL* formula Ψ s.t. $\Phi \equiv \neg\Psi$ (and vice versa)

For each ∀CTL formula Φ there is a ∃CTL formula Ψ s.t. $\Phi \equiv \neg\Psi$ (and vice versa)

Logical characterization of simulation

If s_1 and s_2 are states in a finite TS then the following statements are equivalent:

(1) $s_1 \preceq_{\mathcal{T}} s_2$

(2) for all $\forall$**CTL** formulas Φ:

if $s_2 \models \Phi$ then $s_1 \models \Phi$

(3) for all $\forall$**CTL*** formulas Φ:

if $s_2 \models \Phi$ then $s_1 \models \Phi$

Logical characterization of simulation

If s_1 and s_2 are states in a finite TS then the following statements are equivalent:

(1) $s_1 \preceq_{\mathcal{T}} s_2$

(2∀) for all ∀**CTL** formulas Φ:

$$\text{if } s_2 \models \Phi \text{ then } s_1 \models \Phi$$

(3∀) for all ∀**CTL*** formulas Φ:

$$\text{if } s_2 \models \Phi \text{ then } s_1 \models \Phi$$

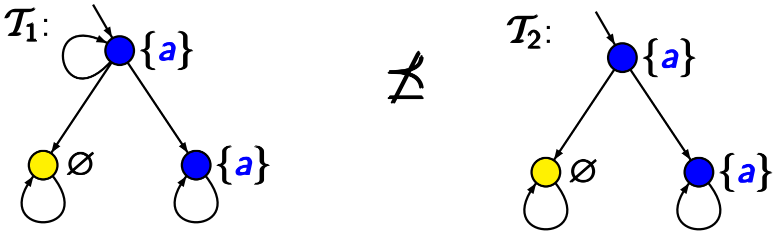
(2∃) for all ∃**CTL** formulas Ψ:

$$\text{if } s_1 \models \Psi \text{ then } s_2 \models \Psi$$

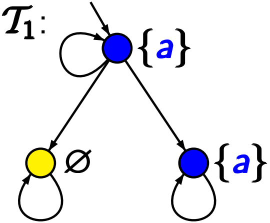
(3∃) for all ∃**CTL** formulas Ψ:

$$\text{if } s_1 \models \Psi \text{ then } s_2 \models \Psi$$

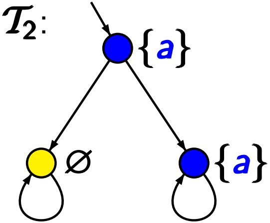
Example: ∀CTL/∃CTL and simulation



Example: ∀CTL/∃CTL and simulation



$\not\preceq$



$\mathcal{T}_1 \not\models \forall\bigcirc(\forall\bigcirc\neg a \vee \forall\bigcirc a)$

$\mathcal{T}_2 \models \forall\bigcirc(\forall\bigcirc\neg a \vee \forall\bigcirc a)$

∀CTL formula

$\mathcal{T}_1 \models \exists\bigcirc(\exists\bigcirc\neg a \wedge \exists\bigcirc a)$

$\mathcal{T}_2 \not\models \exists\bigcirc(\exists\bigcirc\neg a \wedge \exists\bigcirc a)$

∃CTL formula

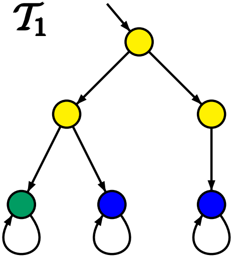
Characterizations of simulation equivalence

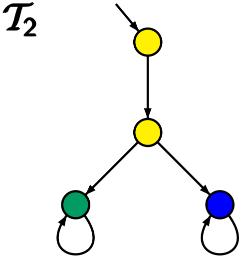
for finite TS without terminal states:

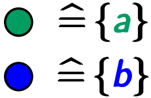
$\mathcal{T}_1 \simeq \mathcal{T}_2$	iff	$\mathcal{T}_1 \preceq \mathcal{T}_2$ and $\mathcal{T}_2 \preceq \mathcal{T}_1$
	iff	$\mathcal{T}_1, \mathcal{T}_2$ satisfy the same $\forall$CTL* formulas
	iff	$\mathcal{T}_1, \mathcal{T}_2$ satisfy the same $\forall$CTL formulas
	iff	$\mathcal{T}_1, \mathcal{T}_2$ satisfy the same $\exists$CTL* formulas
	iff	$\mathcal{T}_1, \mathcal{T}_2$ satisfy the same $\exists$CTL formulas

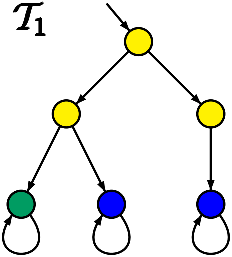
... even holds for $\forall\text{CTL}^*_{\setminus \mathsf{U},\mathsf{W}}$, $\forall\text{CTL}_{\setminus \mathsf{U},\mathsf{W}}$,
$\exists\text{CTL}^*_{\setminus \mathsf{U},\mathsf{W}}$, $\exists\text{CTL}_{\setminus \mathsf{U},\mathsf{W}}$

Simulation equivalence

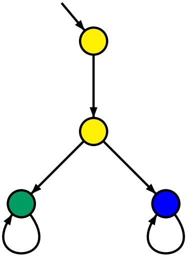






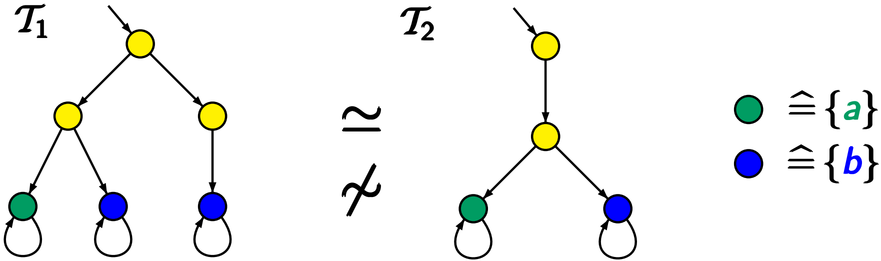


$\simeq$



$\mathcal{T}_1$, $\mathcal{T}_2$ cannot be distinguished by the temporal logics
$\forall$**CTL**, $\forall$**CTL***, $\exists$**CTL**, or $\exists$**CTL***,

Simulation equivalence

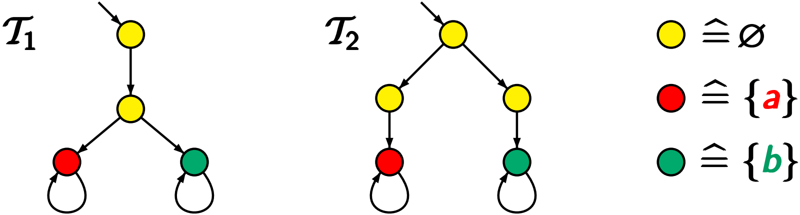


$\mathcal{T}_1$, $\mathcal{T}_2$ cannot be distinguished by the temporal logics **∀CTL**, **∀CTL***, **∃CTL**, or **∃CTL***,

but by **CTL**:

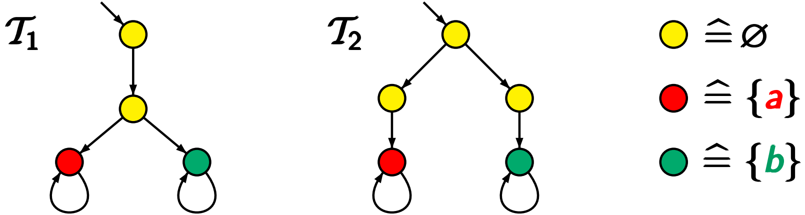
$$\mathcal{T}_1 \not\models \forall\bigcirc(\exists\bigcirc a \wedge \exists\bigcirc b)$$

$$\mathcal{T}_2 \models \forall\bigcirc(\exists\bigcirc a \wedge \exists\bigcirc b)$$



Does there exist a **∃CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

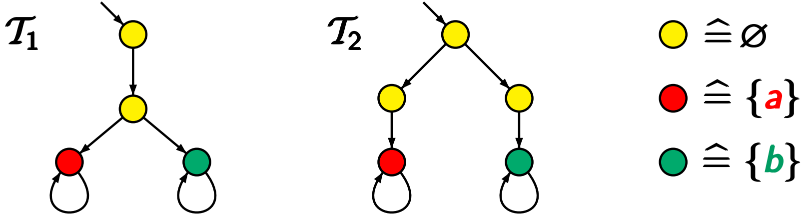
Does there exist ...?



Does there exist a $\exists$**CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

yes, as $\mathcal{T}_1 \not\preceq \mathcal{T}_2$, e.g., $\Phi = \exists\bigcirc(\exists\bigcirc a \ \wedge \ \exists\bigcirc b)$

Does there exist ...?

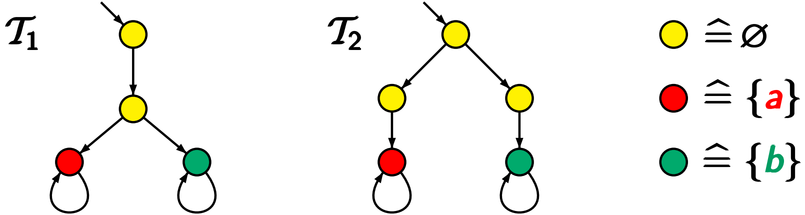


Does there exist a **∃CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

yes, as $\mathcal{T}_1 \not\preceq \mathcal{T}_2$, e.g., $\Phi = \exists\bigcirc(\exists\bigcirc a \ \wedge \ \exists\bigcirc b)$

Does there exist a **∀CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

Does there exist ...?



Does there exist a $\exists$**CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

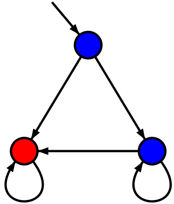
yes, as $\mathcal{T}_1 \not\preceq \mathcal{T}_2$, e.g., $\Phi = \exists\bigcirc(\exists\bigcirc a \ \wedge \ \exists\bigcirc b)$

Does there exist a $\forall$**CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

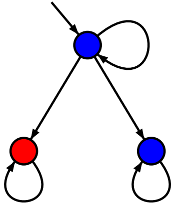
no, as $\mathcal{T}_2 \preceq \mathcal{T}_1$

Does there exist ...?

$\mathcal{T}_1$



$\mathcal{T}_2$



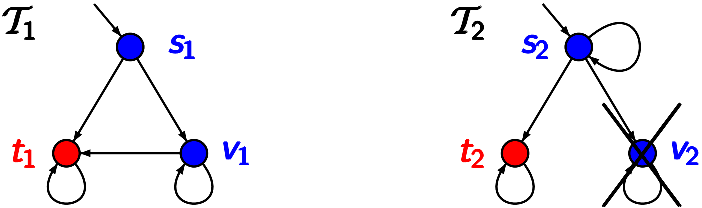
Does there exist a **∃CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

Does there exist ...?

Does there exist a $\exists$**CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

no, since $\mathcal{T}_1 \simeq \mathcal{T}_2$

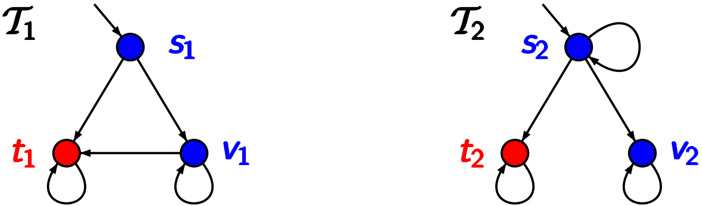
Does there exist ...?



Does there exist a $\exists$**CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

no, since $\mathcal{T}_1 \simeq \mathcal{T}_2$

simulation for $(\mathcal{T}_1, \mathcal{T}_2)$: $\{(s_1, s_2), (v_1, s_2), (t_1, t_2)\}$



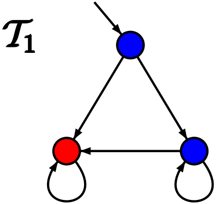
Does there exist a $\exists$**CTL** formula Φ s.t.
$\mathcal{T}_1 \models \Phi$ and $\mathcal{T}_2 \not\models \Phi$?

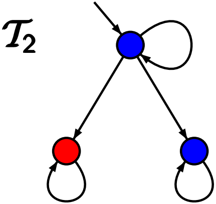
no, since $\mathcal{T}_1 \simeq \mathcal{T}_2$

simulation for $(\mathcal{T}_1, \mathcal{T}_2)$: $\{(s_1, s_2), (v_1, s_2), (t_1, t_2)\}$

simulation for $(\mathcal{T}_2, \mathcal{T}_1)$:
$\{(s_2, s_1), (s_2, v_1), (v_2, v_1), (t_1, t_2)\}$

Does there exist ...?

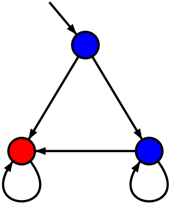




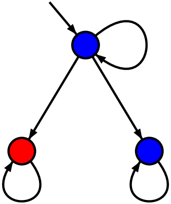
Does there exist a **CTL** formula Φ s.t.
$\mathcal{T}_1 \not\models \Phi$ and $\mathcal{T}_2 \models \Phi$?

Does there exist ...?

$\mathcal{T}_1$



$\mathcal{T}_2$

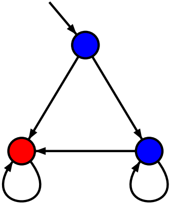


Does there exist a **CTL** formula Φ s.t.
$\mathcal{T}_1 \not\models \Phi$ and $\mathcal{T}_2 \models \Phi$?

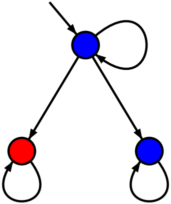
yes, as $\mathcal{T}_1 \not\sim \mathcal{T}_2$, e.g., $\Phi = \exists\bigcirc\forall\Box \textit{blue}$

Does there exist ...?

$\mathcal{T}_1$



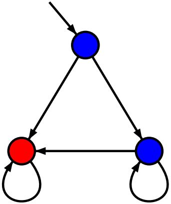
$\mathcal{T}_2$

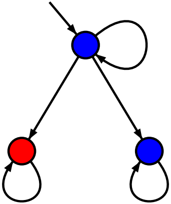


> Does there exist a **CTL** formula Φ s.t.
> $\mathcal{T}_1 \not\models \Phi$ and $\mathcal{T}_2 \models \Phi$?

yes, as $\mathcal{T}_1 \not\sim \mathcal{T}_2$, e.g., $\Phi = \exists\bigcirc\forall\Box \textit{blue}$

> Does there exist a **LTL** formula φ s.t.
> $\mathcal{T}_1 \not\models \varphi$ and $\mathcal{T}_2 \models \varphi$?





Does there exist a **CTL** formula Φ s.t.
$\mathcal{T}_1 \not\models \Phi$ and $\mathcal{T}_2 \models \Phi$?

yes, as $\mathcal{T}_1 \not\sim \mathcal{T}_2$, e.g., $\Phi = \exists\bigcirc\forall\Box \textit{blue}$

Does there exist a **LTL** formula φ s.t.
$\mathcal{T}_1 \not\models \varphi$ and $\mathcal{T}_2 \models \varphi$?

no, as $\mathcal{T}_1$, $\mathcal{T}_2$ are simulation equivalent

Simulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

simulation quotient $\mathcal{T}/\simeq$:

transition system that arises from $\mathcal{T}$ by collapsing all simulation equivalent states

Simulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS. Then:

$$\mathcal{T}/\!\simeq \;\stackrel{\text{def}}{=}\; (S/\!\simeq, Act', \rightarrow_{\simeq}, S_0', AP', L')$$

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS. Then:

$$\mathcal{T}/\simeq \;\stackrel{\text{def}}{=}\; (S/\simeq, Act', \rightarrow_{\simeq}, S_0', AP', L')$$

- state space $S/\simeq$ $\longleftarrow$ $\boxed{\text{set of all simulation equivalence classes}}$

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS. Then:

$$\mathcal{T}/\simeq \;\stackrel{\text{def}}{=}\; (S/\simeq, Act', \rightarrow_{\simeq}, S_0', AP', L')$$

- state space $S/\simeq$ $\longleftarrow$ $\boxed{\text{set of all simulation equivalence classes}}$
- initial states: $S_0' = \{[s] : s \in S_0\}$
- labeling: $AP' = AP$ and $L'([s]) = L(s)$

$$[s] = \{s' \in S : s \simeq_{\mathcal{T}} s'\}$$

Simulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS. Then:

$$\mathcal{T}/\simeq \ \stackrel{\text{def}}{=} \ (S/\simeq, Act', \rightarrow_{\simeq}, S_0', AP', L')$$

- state space $S/\simeq$ $\longleftarrow$ set of all simulation equivalence classes
- initial states: $S_0' = \{[s] : s \in S_0\}$
- labeling: $AP' = AP$ and $L'([s]) = L(s)$
- transition relation: $\dfrac{s \longrightarrow s'}{[s] \longrightarrow_{\simeq} [s']}$

  action labels: irrelevant

Similarity of $\mathcal{T}$ and $\mathcal{T}/{\simeq}$

Similarity of $\mathcal{T}$ and $\mathcal{T}/\simeq$

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS. Then:

$$\mathcal{T}/\simeq \;=\; (S/\simeq, Act', \rightarrow_{\simeq}, S_0', AP, L')$$

where the transitions are given by $\dfrac{s \longrightarrow s'}{[s] \longrightarrow_{\simeq} [s']}$

$\mathcal{T}$ and $\mathcal{T}/\simeq$ are simulation equivalent, i.e.,

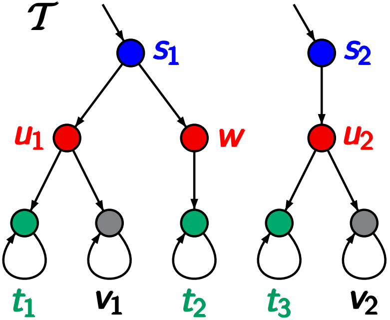
$$\mathcal{T} \preceq \mathcal{T}/\simeq \quad \text{and} \quad \mathcal{T}/\simeq \preceq \mathcal{T}$$

Proof. provide simulations for $(\mathcal{T}, \mathcal{T}/\simeq)$ and $(\mathcal{T}/\simeq, \mathcal{T})$

simulation for $(\mathcal{T}, \mathcal{T}/\simeq)$: $\{(s, [s]) : s \in S\}$

simulation for $(\mathcal{T}/\simeq, \mathcal{T})$: ?

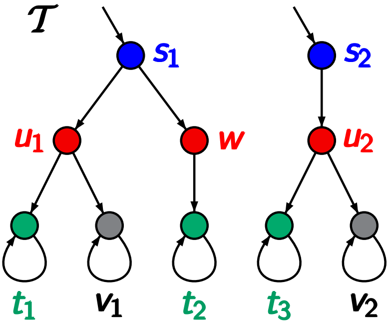
Example: simulation quotient



t_1, t_2, t_3 are simulation equivalent

v_1, v_2 are simulation equivalent

Example: simulation quotient

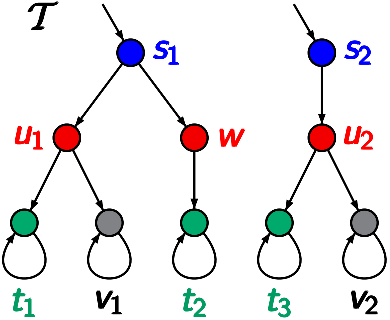


t_1, t_2, t_3 are simulation equivalent

v_1, v_2 are simulation equivalent

$u_1 \simeq u_2$, $\quad w \preceq u_1, u_2$, but $w \not\simeq u_1, u_2$

Example: simulation quotient



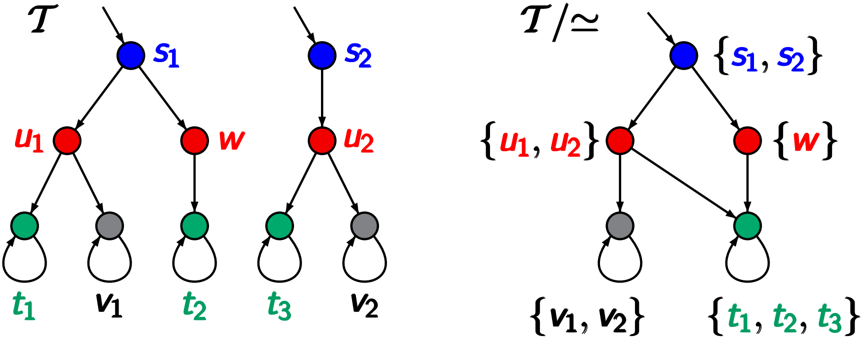
t_1, t_2, t_3 are simulation equivalent

v_1, v_2 are simulation equivalent

$u_1 \simeq u_2$, $\quad w \preceq u_1, u_2$, but $w \not\simeq u_1, u_2$

$s_1 \simeq s_2$

Example: simulation quotient



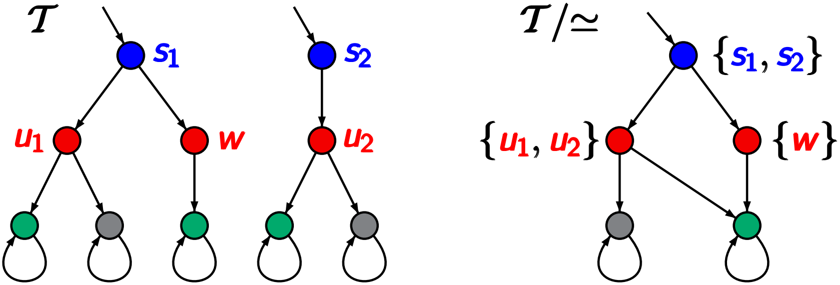
t_1, t_2, t_3 are simulation equivalent

v_1, v_2 are simulation equivalent

$u_1 \simeq u_2$, $\quad w \preceq u_1, u_2$, but $w \not\simeq u_1, u_2$

$s_1 \simeq s_2$

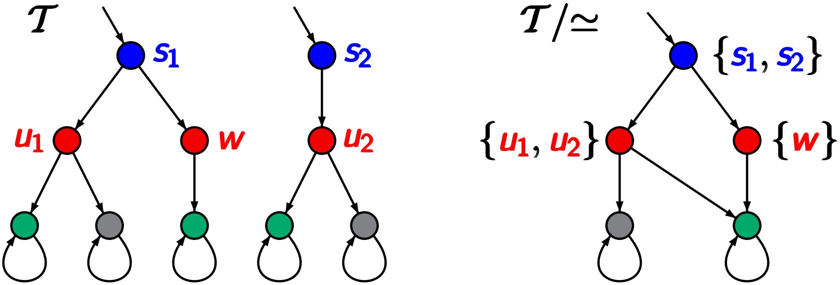
Example: simulation quotient



simulation for $(\mathcal{T}, \mathcal{T}/\!\simeq)$:

$$\big\{ (s, [s]) : s \text{ is a state in } \mathcal{T} \big\}$$

Example: simulation quotient



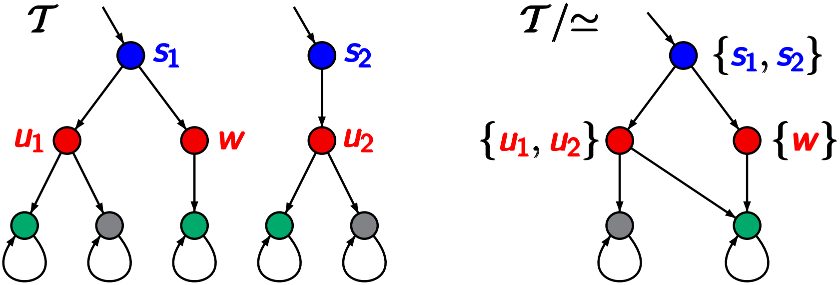
simulation for $(\mathcal{T}, \mathcal{T}/\simeq)$:

$$\big\{ (s, [s]) : s \text{ is a state in } \mathcal{T} \big\}$$

but $\big\{ ([s], s) : s \text{ is a state in } \mathcal{T} \big\}$

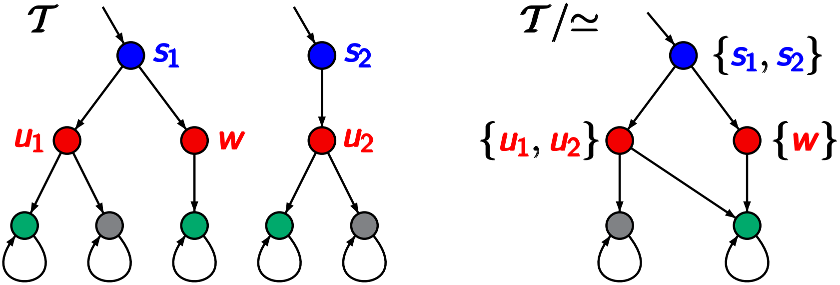
is <u>not</u> a simulation for $(\mathcal{T}/\simeq, \mathcal{T})$

Example: simulation quotient



show that $\mathcal{R} = \{([s], s) : s \text{ is a state in } \mathcal{T}\}$
is <u>not</u> a simulation for $(\mathcal{T}/\simeq, \mathcal{T})$

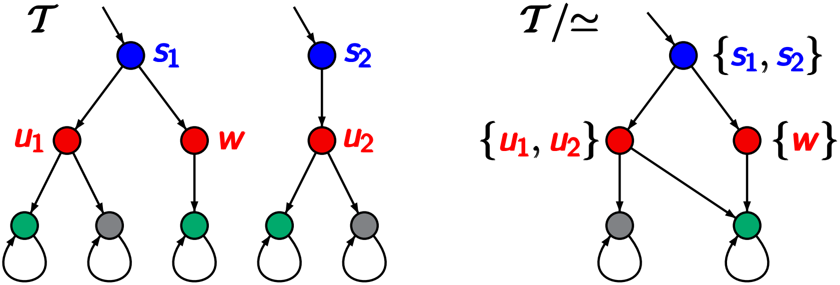
Example: simulation quotient



show that $\mathcal{R} = \{([s], s) : s \text{ is a state in } \mathcal{T}\}$
is <u>not</u> a simulation for $(\mathcal{T}/\simeq, \mathcal{T})$

regard $(\{s_1, s_2\}, s_2) \in \mathcal{R}$ and $\{s_1, s_2\} \rightarrow_{\simeq} \{w\}$

Example: simulation quotient



show that $\mathcal{R} = \{([s], s) : s \text{ is a state in } \mathcal{T}\}$ is not a simulation for $(\mathcal{T}/\simeq, \mathcal{T})$

regard $(\{s_1, s_2\}, s_2) \in \mathcal{R}$ and $\{s_1, s_2\} \to_{\simeq} \{w\}$

there is no transition $s_2 \to w'$ in $\mathcal{T}$ s.t. $(\{w\}, w') \in \mathcal{R}$

Similarity of $\mathcal{T}$ and $\mathcal{T}/{\simeq}$

Let $\mathcal{T} = (S, \mathit{Act}, \rightarrow, S_0, AP, L)$ be a TS. Then:

$$\mathcal{T}/{\simeq} \;=\; (S/{\simeq}, \mathit{Act}', \rightarrow_{\simeq}, S_0', AP, L')$$

where the transitions are given by $\dfrac{s \longrightarrow s'}{[s] \longrightarrow_{\simeq} [s']}$

> $\mathcal{T}$ and $\mathcal{T}/{\simeq}$ are simulation equivalent, i.e.,
> $\mathcal{T} \preceq \mathcal{T}/{\simeq}$ and $\mathcal{T}/{\simeq} \preceq \mathcal{T}$

Proof. provide simulations for $(\mathcal{T}, \mathcal{T}/{\simeq})$ and $(\mathcal{T}/{\simeq}, \mathcal{T})$

simulation for $(\mathcal{T}, \mathcal{T}/{\simeq})$: $\{(s, [s]) : s \in S\}$

simulation for $(\mathcal{T}/{\simeq}, \mathcal{T})$: **?**

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS. Then:

$$\mathcal{T}/\simeq \;=\; (S/\simeq, Act', \rightarrow_{\simeq}, S_0', AP, L')$$

where the transitions are given by $\dfrac{s \longrightarrow s'}{[s] \longrightarrow_{\simeq} [s']}$

$\mathcal{T}$ and $\mathcal{T}/\simeq$ are simulation equivalent, i.e.,
$\mathcal{T} \preceq \mathcal{T}/\simeq$ and $\mathcal{T}/\simeq \preceq \mathcal{T}$

Proof. provide simulations for $(\mathcal{T}, \mathcal{T}/\simeq)$ and $(\mathcal{T}/\simeq, \mathcal{T})$

simulation for $(\mathcal{T}, \mathcal{T}/\simeq)$: $\{(s, [s]) : s \in S\}$

simulation for $(\mathcal{T}/\simeq, \mathcal{T})$: $\{([s], t) : s \preceq_{\mathcal{T}} t\}$